



**Verónica de Jesus
Branco Pacheco
da Rocha**

**Sistema de Informação de Suporte à Avaliação
Multidimensional da Funcionalidade de Pessoas
Idosas**



**Verónica de Jesus
Branco Pacheco
da Rocha**

**Sistema de Informação de Suporte à Avaliação
Multidimensional da Funcionalidade de Pessoas
Idosas**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor Joaquim Manuel Henriques de Sousa Pinto, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor José Manuel Matos Moreira

Professor Auxiliar da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Professor Doutor Fernando Joaquim Lopes Moreira

Professor Associado da Universidade Portucalense

Professor Doutor Joaquim Manuel Henriques de Sousa Pinto

Professor Auxiliar da Universidade de Aveiro

**agradecimentos /
acknowledgements**

Na realização da presente dissertação, contei com o apoio direto ou indireto de múltiplas pessoas às quais estou profundamente grata.

Ao meu orientador, professor Joaquim Sousa Pinto, pela orientação prestada, pelo seu incentivo, disponibilidade e apoio demonstrado.

À minha avó, Maria da Conceição, que já não se encontra entre nós, cuja convivência contribuiu para o meu ávido interesse pelo tema desta dissertação. Um agradecimento especial por ser sempre a minha maior fã e o meu porto-seguro.

Aos meus pais, pelo seu apoio incansável e constante, pela partilha de conhecimento e sabedoria, pela disponibilidade e incentivo ao meu trabalho.

Aos meus amigos, em especial à Mariana e à Maria João, por estarem ao meu lado nos bons e maus momentos, pela mão sempre estendida, pelo sorriso desinibido e pela amizade incondicional.

palavras-chave

Avaliação Multidimensional da Funcionalidade, Pessoas Idosas, Sistema de Gestão de Inquéritos, Instrumentos de Avaliação.

resumo

A funcionalidade humana, que se define como a capacidade de uma pessoa realizar as atividades da sua vida diária com autonomia e independência, bem como ter uma participação ativa na sociedade que o rodeia, está condicionada por vários fatores que têm impacto na qualidade de vida das pessoas. Existem vários modelos teóricos que explicam a funcionalidade humana, no entanto, a sua complexidade inerente tem dificultado a sua aplicação, de modo que não existe um modelo universal para a avaliação da funcionalidade humana. Para a avaliação das diferentes dimensões de funcionalidade existe um conjunto diversificado de instrumentos adequados. O objetivo desta dissertação consiste no desenvolvimento de um sistema de informação que apoie a avaliação multidimensional da funcionalidade de pessoas idosas, através do registo e análise da aplicação dos instrumentos escolhidos pelo prestador de serviços. O sistema desenvolvido consiste numa plataforma web que permite suportar um conjunto abrangente de instrumentos vocacionados quer para a avaliação global, quer para dimensões específicas, possibilitando ao profissional a seleção dos instrumentos mais adequados para um dado paciente e registo das avaliações efetuadas. Os resultados da avaliação serão apresentados de modo conciso e consistente, permitindo ao prestador de cuidados aceder a toda a informação necessária para a tomada de decisão relativa às intervenções que devem ser realizadas dado o estado do paciente.

keywords

Multimentional Functionality Evaluation, Elderly People, Inquiry Management System, Evaluation Instruments.

abstract

Human functionality is defined as a person's capacity to perform daily activities autonomously and independently, as well as be a part of their surrounding society. It is conditioned by various factors that impact quality of life. There are many theoretical models that intend to explain human functionality, however, its complexity results in a difficult application of these models. As a result, there is no universal model to use in human functionality evaluation. There are, however, different dimensions to it that can be evaluated through a set of diverse suitable instruments. The purpose of this dissertation is to develop an information system that supports the multidimensional functionality evaluation of elderly people through the registry and analysis of the application and result of instruments chosen by the care provider. The developed system consists in a web platform that supports a comprehensive set of instruments aimed either to global evaluation or to specific dimensions, allowing the selection of the most adequate instruments, chosen by the professional care giver, that suits best the patient's condition. The results obtained are presented in a concise and amicable way so that allows the care giver all the essential information for a decision making related to the patient's state.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	3
1.3 Método de Desenvolvimento de Software	3
1.4 Organização do Documento	4
2 Estado de Arte	5
2.1 Inquéritos de um Sistema de Apoio à Avaliação Multidimensional da Funcionalidade de Pessoas Idosas	5
2.2 Sistemas de Gestão de Inquéritos	10
2.2.1 Survio	10
2.2.2 SurveyMonkey	12
2.2.3 QuestionPro	14
2.2.4 Análise Comparativa	15
2.3 Base de Dados	16
2.3.1 Relacional vs. Não Relacional	16
2.3.2 Base de Dados Relacional	17
2.3.3 Base de Dados Não Relacional	19
2.3.4 Comparação entre DBMS	19
2.4 Arquiteturas	20
2.4.1 Arquitetura Model-View-Controller	20
2.4.2 Arquitetura Model-View-Presenter	21
2.4.3 Arquitetura Model-View-ViewModel	22
2.4.4 Comparação entre Arquiteturas	23
2.5 Frameworks Web	23
2.5.1 ASP.NET Model View Controller	23
2.5.2 Django	26

2.5.3	Ambiente de Desenvolvimento	27
3	Arquitetura	28
3.1	Contexto	28
3.2	Requisitos Funcionais	28
3.3	Requisitos não Funcionais	30
3.4	Perfis de Utilização	31
3.5	Casos de Uso	32
3.6	Mockups	42
3.7	Conclusão	47
4	Implementação	48
4.1	Tecnologias Utilizadas	48
4.2	Arquitetura do Sistema	49
4.3	Base de Dados	50
4.4	Aplicação Web	51
4.4.1	Object-Relational Mapping	51
4.4.2	Modelo	52
4.4.3	Controlador	54
4.4.4	View	57
4.4.5	Integração	68
5	Conclusão e Trabalho Futuro	69
5.1	Conclusão	69
5.2	Trabalho Futuro	70
	Bibliografia	71

Lista de Figuras

2.1	Questionário PHQ-9 [1]	9
2.2	Questionário MoCA [2]	10
2.3	Modelos de questionários presentes no Survio [3]	11
2.4	Escolha de um tipo de pergunta num questionário feito à medida [3]	12
2.5	Criação de um questionário no SurveyMonkey [4]	13
2.6	Criação de um questionário no QuestionPro [5]	14
2.7	Instância e Base de Dados Oracle [6]	18
2.8	Diagrama de relação entre os componentes do MVC	21
2.9	MVP Passive View [7]	22
2.10	MVP Supervising Presenter [7]	22
2.11	Diagrama de relação entre os componentes do MVMM [7]	22
2.12	Arquitetura MTV [8]	26
3.1	Diagrama de casos de uso do Utilizador comum	32
3.2	Diagrama de casos de uso do Administrador	33
3.3	Mockup de uma página de criação de um novo instrumento	43
3.4	Mockup de uma página de edição de instrumentos	43
3.5	Mockup de uma página de remoção de um instrumento	44
3.6	Mensagem de erro que aparece quando o administrador tenta remover um instrumento com avaliações associadas	44
3.7	Mockup de uma página que enumera todos os instrumentos e permite procurar um instrumento específico	45
3.8	Mockup de uma página que permite ver os detalhes de um instrumento	46
3.9	Mockup da página de detalhes de um utente	47
4.1	Arquitetura do Sistema	49
4.2	Diagrama da BD	50
4.3	Modelo de Utilizador da Aplicação	53
4.4	Modelo da avaliação	54
4.5	Modelos desenvolvidos à medida	55
4.6	Query para extrair o objeto "Question" da BD	55
4.7	Query para obter o objeto "Entry" pretendido da BD e eliminá-lo da mesma	55
4.8	Query para obter o inteiro guardado no campo valueID da tabela AnswerValues	56

4.9	Função "SaveChanges" das avaliações	56
4.10	Barra de Navegação do Administrador	57
4.11	Menu de Gestão de Tabelas	58
4.12	Menu de Gestão de Utilizadores	58
4.13	Página que contém a lista de instrumentos	59
4.14	Página de criação de instrumentos	59
4.15	Página de eliminação do instrumento Índice de Pfeffer	60
4.16	Página "Ver Detalhes" do instrumento Índice de Pfeffer	61
4.17	Página "Ver Detalhes" do instrumento Escala de Barthel	61
4.18	Página de criação de uma nova questão do Índice de Pfeffer	62
4.19	Página de detalhes da instituição Hospital da Luz Aveiro	62
4.20	Barra de navegação do utilizador	63
4.21	Menu de Gestão de Avaliações	63
4.22	Menu de Gestão de Utentes	63
4.23	Página para adicionar uma nova avaliação	64
4.24	Página para submeter as respostas da avaliação selecionada	64
4.25	Página de detalhes de uma avaliação parte 1	65
4.26	Página de detalhes de uma avaliação parte 2	65
4.27	Detalhes de um utente - Dados pessoais, Contacto e Avaliações efetuadas .	66
4.28	Detalhes de um utente - Estatísticas do utente por instrumento	67
4.29	Código javascript e html com C Sharp embutido para criação dinâmica de um gráfico	67

Lista de Tabelas

2.1	Tabela de Comparação de Sistemas de Gestão de Inquéritos	16
3.1	Tabela de Descrição dos casos de uso	42

Capítulo 1

Introdução

1.1 Motivação

O índice de envelhecimento da população portuguesa aumenta de ano para ano e, de acordo com as projeções, o incremento substancial da proporção de idosos continuará no futuro [9]. Tal resulta da diminuição da taxa de natalidade e do aumento da esperança de vida e implica a necessidade de garantia de qualidade de vida (QdV) à medida que os indivíduos envelhecem.

QdV, uma expressão que nos dias de hoje é utilizada e explorada por diversos sectores para fins igualmente diversificados, continua a ser um conceito subjetivo. Por isso, a Organização Mundial de Saúde (OMS) tem vindo a desenvolver diversos estudos para estabelecer um referencial conceptual da QdV, o que é imperativo para a operacionalização de métodos de avaliação objetivos [10].

Independentemente de qual a perspectiva que se considera para a avaliação da QdV de uma pessoa, um dos aspetos fundamentais é a sua funcionalidade, entendida como a capacidade dessa pessoa para realizar as suas atividades de vida diária (por exemplo, alimentar-se ou cuidar da sua higiene pessoal) com autonomia e independência, bem como ter uma participação ativa na sociedade que a rodeia.

A funcionalidade humana está condicionada por vários fatores que podem induzir o aparecimento de incapacidades que, conseqüentemente, têm impacto na QdV das pessoas. Dentro desses fatores destacam-se: deficiência do foro físico ou mental; depressão; problemas cognitivos (por exemplo, comprometimento da memória ou da função executiva); reduzida frequência de contactos sociais; baixa atividade física; índice de massa corporal elevado ou baixo; comorbidade de várias doenças; incorreta auto-avaliação do estado de saúde; incorreta autoavaliação da função visual; e consumo exagerado de álcool.

Segundo a OMS [11], as nove causas de incapacidade entre a população idosa são doenças cardiovasculares, problemas músculo-esqueléticos, hipertensão, acidentes vasculares cerebrais (AVC), diabetes, doenças oncológicas, doença pulmonar obstrutiva crónica, patologias mental (principalmente demência e depressão) e cegueira ou deficiência visual. Conseqüentemente, aproximadamente 25% das pessoas com idade superior a 65 anos necessitam de

ajuda para a realização das atividades do dia-a-dia. No caso das pessoas idosas com idade superior a 85 anos, a taxa chega a atingir os 50%.

Existem diversos modelos teóricos que explicam a funcionalidade humana, sendo estes o modelo de Incapacidade de Nagi; a Classificação Internacional de Deficiências, Incapacidades e Desvantagens; o processo de produção de desvantagem; e a Classificação Internacional de Funcionalidade, Incapacidade e Saúde.

Apesar da riqueza conceptual da Classificação Internacional de Funcionalidade, Incapacidade e Saúde, também conhecida por CIF, a respetiva complexidade tem dificultado a sua operacionalização. Por isso, atualmente, não há nenhum modelo de funcionalidade humana que seja o adotado universalmente. Acresce-se que para a avaliação global da funcionalidade e para a avaliação das diferentes dimensões de funcionalidade (ou seja, dimensões física, mental, emocional e social) existem um conjunto diversificados de instrumentos como, por exemplo:

- O índice de Independência nas Atividades de Vida Diária de Katz [12];
- O índice de Barthel [13];
- A escala de Lawton das Atividades Instrumentais de Vida Diária [14];
- O índice de Pfeffer [15];
- Timed get up and go [16];
- A escala de equilíbrio de Berg [17];
- O Tinetti Performance-Oriented Mobility Assessment (POMA) [18];
- A Morse Fall Scale (MFS) [19];
- A Mini-Mental Statement Examination (MMSE) [20];
- O Short Portable Mental Status Questionnaire (SPMSQ) [21];
- O Teste do Desenho do Relógio [22];
- O Teste de Fluência Verbal e Semântica [23];
- A Montreal Cognitive Assessment (MoCA) [2];
- A Geriatric Depression Scale (GDS) [24];
- O Patient Health Questionnaire (PHQ-9) [25];
- A Lubben Social Network Scale (LSNS) [26];
- A MOS Social Support Survey [27];
- A Escala de Satisfação com o Suporte Social (ESSS) [28];

- A Medida de Independência Funcional (MIF) [29];
- A 36-Item Short Form Survey (SF- 36) [30];
- O WHOQOL-BREF - Brief version of World Health Organization Quality of Life Questionnaire [31];
- O Older Americans Resources and Services (OARS) [32];
- A World Health Organization Disability Assessment Schedule (WHODAS 2.0) [33].

Em 2012, um painel de especialistas em cuidados a prestar a idosos com múltiplas patologias, recomendou que qualquer prestação de cuidados estruturada deve incluir obrigatoriamente uma avaliação multidimensional contemplando sintomas, funções físicas e cognitivas, depressão, ansiedade e atividades de vida diária [34].

No entanto, apesar de diversos instrumentos já serem utilizados para a avaliação das diversas dimensões da funcionalidade humana, não existe qualquer plataforma que permita o armazenamento e visualização dos seus resultados.

Normalmente, o resultado das avaliações efetuadas são registadas em papel, o que apresenta diversos problemas, nomeadamente dificuldade de leitura, dificuldade de partilha ou dificuldade de obtenção de resultados agregados.

1.2 Objectivos

O objetivo da presente dissertação é o desenvolvimento de um sistema de informação que apoie a avaliação multidimensional da funcionalidade de pessoas idosas através do registo e análise da aplicação dos instrumentos adequados.

Para tal o sistema deverá: i) suportar um conjunto abrangente de instrumentos vocacionados quer para a avaliação global, quer para a avaliação das dimensões física, mental, emocional e social da funcionalidade da pessoa idosa; ii) possibilitar ao prestador de cuidados a seleção dos instrumentos mais adequados para a avaliação multidimensional da funcionalidade de uma determinada pessoa idosa; iii) registar todas as avaliações efetuadas a uma determinada pessoa idosa; iv) providenciar mecanismos de acesso a todos os registos relativos a uma determinada pessoa idosa; v) providenciar ao prestador de cuidados um Dashboard que integre os resultados da avaliação global e da avaliação das várias dimensões da funcionalidade humana; e vi) permitir o prestador de cuidados aceda de uma forma amigável a toda a informação necessária para a tomada de decisão relativa às intervenções que devem ser realizadas.

1.3 Método de Desenvolvimento de Software

No âmbito da presente dissertação, o desenvolvimento de software baseia-se no método AGILE, ou ágil, que é conhecido por providenciar uma estrutura concetual para orientação

de projetos de engenharia de software. Este método tenta minimizar o risco pelo desenvolvimento de software através de procedimentos que permitem melhorias nas diferentes fases do projecto, o que possibilita que os resultados do desenvolvimento possam satisfazer os requisitos que foram propostos [35].

De acordo com este método, o projeto de desenvolvimento de um sistema de informação é composto por sete passos: Conceção, Iniciação, Análise de Requisitos, Design, Construção, Teste e Produção.

O primeiro passo é a concepção do produto, que tem como maior objetivo que a ideia do produto seja bem definida. Este passo é muito importante para que as fases seguintes sejam bem sucedidas. A inicialização e a análise podem ocorrer em simultâneo e consistem na análise de requisitos, definição de casos de uso e aplicação das ideias desenvolvidas na fase de concepção. Só após a conclusão destas duas fases é que se segue o design e construção, que podem ser executadas em paralelo. O modelo da aplicação tem que estar definido antes da fase de construção, no entanto, alguns detalhes podem vir a ser alterados durante o desenvolvimento de modo a aprimorar a aplicação. Por fim, existem as fases de teste e produção, que são sequenciais, pois só após todos os testes terem sido bem sucedidos é que se considera que o produto está pronto para produção.

1.4 Organização do Documento

O presente documento contém a descrição de toda a pesquisa, análise, desenvolvimento e conclusões relacionados com o trabalho desenvolvido com o fim de cumprir os objetivos propostos de dissertação. Neste sentido, existem cinco capítulos:

- O capítulo atual, Introdução, apresenta o contexto, motivação, objetivos e metodologias utilizadas nesta dissertação.
- O segundo capítulo, Estado de Arte, apresenta toda a pesquisa executada, bem como os seus resultados, a nível de base de dados, frameworks e tecnologias utilizadas em sistemas de informação e a nível de sistemas de gestão de inquéritos já existentes.
- O terceiro capítulo, Arquitetura, descreve o funcionamento esperado do sistema através de casos de uso e descrição de requisitos, funcionais e não funcionais, a ele associados.
- O quarto capítulo, Implementação, apresenta a arquitetura geral e tecnologias utilizadas, o diagrama da base de dados subjacente e descreve os passos presentes no desenvolvimento do sistema, bem como os problemas, soluções e decisões associadas a esse processo.
- O quinto e último capítulo, Conclusões e Trabalho Futuro, apresenta as considerações finais acerca do trabalho desenvolvido e sugere trabalho adicional para o futuro.

Capítulo 2

Estado de Arte

O presente capítulo foca-se nos sistemas de gestão de inquéritos, referindo algumas plataformas já existentes, estabelecendo uma análise comparativa e verificando a sua adaptabilidade para suportar os diversos instrumentos, quer na avaliação global da funcionalidade, quer para a avaliação das diferentes dimensões de funcionalidade.

Adicionalmente, apresentam-se tecnologias de Base de Dados, Arquiteturas de Sistemas de Informação e web frameworks.

2.1 Inquéritos de um Sistema de Apoio à Avaliação Multidimensional da Funcionalidade de Pessoas Idosas

Como já foi referido anteriormente, para a avaliação global da funcionalidade existe um conjunto diversificado de instrumentos. Em adição, existem outros conjuntos específicos para as diferentes dimensões ou categorias de funcionalidade, sendo estas física, mental, emocional e social.

No caso da categoria física destacam-se os seguintes instrumentos:

- **Índice de Barthel**

Avaliação das atividades básicas da vida diária na admissão de doentes no hospital e após a alta. Avalia dez itens, sendo eles alimentação, higiene pessoal, rotinas diárias, vestir, controlo fecal, controlo vesical, utilização da sanita, transferências cama-cadeira, mobilidade em superfície plana horizontal e subir/descer escadas [13].

- **Escala de Lawton das Atividades Instrumentais de Vida Diária**

Instrumento adequado para avaliar a realização de Atividades Instrumentais de Vida Diária (AIVD), nomeadamente capacidade de utilizar o telefone, fazer compras, preparar refeições, realizar tarefas domésticas, utilizar transportes, ser responsável pela medicação, tratar da roupa e lidar com as finanças [14].

- **Índice de Pfeffer**

Constituído por itens que classificam a capacidade do indivíduo para realizar AIVD e funções cognitivas/sociais como fazer compras, preparar refeições, manter-se atualizado, prestar atenção a programas de rádio ou TV e discuti-los [15].

- **Timed get up and go**

Fundamental para prever o grau de independência para locomoção no meio onde a pessoa vive e para avaliar o risco de quedas e complicações inerentes.

- **Escala de Equilíbrio de Berg**

A sua aplicação envolve a realização de tarefas (e.g. alcançar, girar, transferir-se, permanecer de pé e levantar-se) que permitam a avaliação quer do equilíbrio estático, quer do equilíbrio dinâmico [17].

- **Tinetti Performance-Oriented Mobility Assessment (POMA)**

Avalia o equilíbrio e a marcha em idosos através de um teste de desempenho. A versão mais utilizada do POMA é constituída por 16 itens, nove relativos a manobras de equilíbrio em destabilização do centro de gravidade e sete relativos a manobras de marcha através de atividades sequenciais num circuito de marcha [18].

- **Morse Fall Scale (MFS)**

Avalia qual a probabilidade de uma pessoa poder cair. Esta escala encontra-se organizada em seis parâmetros: histórico de quedas nos últimos três meses, diagnóstico secundário, ajuda na mobilização, terapia endovenosa, marcha e estado mental [19].

Apesar de todos os instrumentos serem adequados à avaliação de funcionalidade na dimensão física, o Índice de Barthel é um dos mais utilizados para a avaliação funcional geriátrica e é um comparador padrão para avaliação das propriedades psicométricas de instrumentos semelhantes.

A nível de funcionalidade mental, existem os seguintes questionários:

- **Mini-Mental State Examination (MMSE)**

Avalia os seguintes domínios: orientação, retenção, atenção e cálculo, evocação, linguagem e habilidade construtiva [20].

- **Short Portable Mental Status Questionnaire (SPMSQ)**

Dez questões que avaliam o défice da função cognitiva ou deterioração intelectual sendo a maioria dos seus itens versões modificadas de outras escalas [21].

- **Teste de Desenho do Relógio**

Avalia, além da componente verbal, aspetos não-verbais e envolve várias áreas da cognição, nomeadamente compreensão verbal do comando, planeamento, memória visual, habilidade visuo-espacial, programação e execução motoras, conhecimento numérico e pensamento abstrato [22].

- **Teste de Fluência Verbal e Semântica**

Avaliação de categorias semânticas pré-definidas e pretende avaliar capacidades como a memória semântica, a recuperação de informação guardada na memória, a organização do pensamento e as estratégias utilizadas para a pesquisa de palavras [23].

- **Montreal Cognitive Assessment (MoCA)**

Avalia diferentes domínios cognitivos, nomeadamente a função executiva, a capacidade visuo-espacial, a memória, a atenção, a concentração, a memória de trabalho, a linguagem e a orientação temporal e espacial [2].

O MMSE é um instrumento amplamente utilizado na prática clínica, devido ao facto de ser facilmente aplicado. Por outro lado, o SPMSQ é específico para utentes com um nível de demência entre moderada a grave. No entanto, pode-se considerar que, nesta categoria, o MoCA é um dos instrumentos de rastreio cognitivo mais usados nos protocolos de avaliação de diversos grupos clínicos, nomeadamente declínio cognitivo ligeiro, doença de Alzheimer ou demência vascular, entre outros.

Os dois instrumentos mais utilizados e conhecidos a nível de avaliação emocional são os seguintes:

- **Escala Geriátrica de Depressão**

Escala de auto-preenchimento composta por 28 itens que avaliam como a pessoa idosa se sentiu na última semana em termos de depressão [24].

- **Patient Health Questionnaire (PHQ-9)**

O instrumento é constituído por nove questões que avaliam a frequência de sintomas depressivos de acordo com um índice de gravidade [25].

Para a dimensão social são usados predominantemente dois instrumentos:

- **Escala de Redes Sociais de Lubben**

Avalia a integração social e o risco de isolamento social em idosos residentes na comunidade [26]. Tem uma versão reduzida e outra expandida.

- **Escala de Suporte Social de MOS**

É constituída por 15 itens de auto-preenchimento e pretende avaliar a satisfação da pessoa relativamente ao suporte social existente [27].

Por fim, é possível a avaliação da funcionalidade geral de um utente utilizando instrumentos multi-facetados como os seguintes:

- **Medida de Independência Funcional (MIF)**

Trata-se de um instrumento de avaliação constituído por 18 itens que se podem dividir em duas áreas: motora (agrega 13 itens referentes a auto-cuidados, controle dos esfíncteres, mobilidade ou transferências e locomoção) e cognitiva (agrega cinco itens referentes a comunicação e cognição social) [29].

- **WHOQOL-BREF - Brief version of World Health Organization Quality of Life Questionnaire**

Constituído por 26 questões organizadas em quatro domínios de qualidade de vida: físico, psicológico, relações sociais e ambiente [31].

- **Older Americans Resources and Services (OARS)**

Composto por cinco sub-escalas, sendo elas recursos sociais, recursos económicos, saúde mental, saúde física, AVD e utilização de serviços [32].

- **World Health Organization Disability Assessment Schedule (WHODAS 2.0)**

Avalia a funcionalidade do indivíduo em seis domínios de atividade: cognição (seis itens), mobilidade (cinco itens); auto-cuidado (quatro itens), relações interpessoais (cinco itens), atividades diárias (oito itens) e participação (oito itens) [33].

No caso destes instrumentos, MIF é muito utilizado em reabilitação para a avaliação do estado funcional de um indivíduo e o WHODAS 2.0 é usado maioritariamente na avaliação da incapacidade resultante da doença e da interação da pessoa com o ambiente.

Na sua forma mais simples, cada um destes instrumentos apresenta uma série de perguntas, as quais devem ser respondidas com base num conjunto normalizado de respostas e igual para todas as perguntas

A título exemplificativo apresenta-se na Figura. 2.1 o instrumento PHQ-9.

O PHQ-9 é composto por perguntas relativas a problemas que o avaliado possa ter tido nas duas semanas anteriores ao momento de avaliação:

1. Pouco interesse em fazer as coisas;
2. Desânimo, desalento ou falta de esperança;
3. Problemas de sono;
4. Cansaço ou falta de energia;
5. Falta ou excesso de apetite;
6. Sentir que não gosta de si próprio ou que é um falhado ou que é uma desilusão;
7. Dificuldade a nível de concentração;
8. Mudanças de comportamento a nível de agitação ou falta dela;
9. Sentir que devia estar morto ou magoar-se a si mesmo.

A resposta a qualquer uma das perguntas anteriores pode ser "Nunca", "Em vários dias", "Em mais de metade do número de dias" ou "Em quase todos os dias", correspondendo, respetivamente a 0, 1, 2 e 3 pontos. O resultado final, ou score total, é o somatório dos pontos associados às respostas das diferentes perguntas. Considerando-se que o avaliado

QUESTIONÁRIO SOBRE A SAÚDE DO PACIENTE-9 (P H Q - 9)				
Durante os últimos 14 dias, em quantos foi afectado/a por algum dos seguintes problemas? (Utilize "✓" para indicar a sua resposta)	Nunca	Em vários dias	Em mais de metade do número de dias	Em quase todos os dias
1. Tive pouco interesse ou prazer em fazer coisas	0	1	2	3
2. Senti desânimo, desalento ou falta de esperança	0	1	2	3
3. Tive dificuldade em adormecer ou em dormir sem interrupções, ou dormi demais	0	1	2	3
4. Senti cansaço ou falta de energia	0	1	2	3
5. Tive falta ou excesso de apetite	0	1	2	3
6. Senti que não gosto de mim próprio/a — ou que sou um(a) falhado/a ou me desiludi a mim próprio/a ou à minha família	0	1	2	3
7. Tive dificuldade em concentrar-me nas coisas, como ao ler o jornal ou ver televisão	0	1	2	3
8. Movimentei-me ou falei tão lentamente que outras pessoas poderão ter notado. Ou o oposto: estive agitado/a a ponto de andar de um lado para o outro muito mais do que é habitual	0	1	2	3
9. Pensei que seria melhor estar morto/a, ou em magoar-me a mim próprio/a de alguma forma	0	1	2	3

FOR OFFICE CODING 0 + + +
=Total Score:

Se indicou alguns problemas, até que ponto é que eles dificultaram o seu trabalho, o cuidar da casa ou o lidar com outras pessoas?

Não dificultaram	Dificultaram um pouco	Dificultaram muito	Dificultaram extremamente
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 2.1: Questionário PHQ-9 [1]

não tem depressão se o score for inferior a quatro, terá depressão ligeira para um score entre cinco e nove, terá depressão moderadamente severa para um score entre 15 e 19, e terá uma depressão severa para um score superior a 20.

Adicionalmente, existe ainda uma outra pergunta que não é considerada para o score e que serve para identificar o grau de dificuldade (nenhuma, alguma, muita ou extrema) que os problemas assinalados nas restantes nove perguntas causaram nas atividades de vida diária do avaliado.

No entanto, outros instrumentos poderão ter maior complexidade e poderão não possuir um conjunto normalizado de respostas, ou seja, o leque de respostas possíveis varia de acordo com o item a avaliar. É o caso do MoCA, apresentado na Figura 2.2, que implica que diferentes tipos de informação tenham que ser registados.

VISUO-ESPACIAL / EXECUTIVA				Copiar o cubo 		Desenhar um Relógio (onze e dez) (3 pontos)		Pontos
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		___/5
NOMEAÇÃO								___/3
MEMÓRIA		Leia a lista de palavras. O sujeito deve repeti-la. Realize dois ensaios. Solicite a evocação da lista 5 minutos mais tarde.		Boca Linho Igreja Cravo Azul		Sem Pontuação		___/5
ATENÇÃO		Leia a sequência de números. (1 número/segundo)		O sujeito deve repetir a sequência. <input type="checkbox"/> 2 1 8 5 4 O sujeito deve repetir a sequência na ordem inversa. <input type="checkbox"/> 7 4 2		___/2		
		Leia a série de letras (1 letra/segundo). O sujeito deve bater com a mão cada vez que for dita a letra A. Não se atribuem pontos se ≥ 2 erros.		<input type="checkbox"/> FBACMNAAJKLBAFAKDEAAAJAMOFAB		___/1		
		Subtrair de 7 em 7 começando em 100. <input type="checkbox"/> 93 <input type="checkbox"/> 86 <input type="checkbox"/> 79 <input type="checkbox"/> 72 <input type="checkbox"/> 65		4 ou 5 subtrações corretas: 3 pontos; 2 ou 3 corretas: 2 pontos; 1 correta: 1 ponto; 0 corretas: 0 pontos		___/3		
LINGUAGEM		Repetir: Eu só sei que hoje devemos ajudar o João. <input type="checkbox"/> O gato esconde-se sempre que os cães entram na sala. <input type="checkbox"/>		___/2				
		Fluência verbal: Dizer o maior número possível de palavras que comecem pela letra "P" (1 minuto). <input type="checkbox"/> _____ (N ≥ 11 palavras)		___/1				
ABSTRAÇÃO		Semelhança p.ex. entre banana e laranja = fruta <input type="checkbox"/> comboio - bicicleta <input type="checkbox"/> relógio - régua		___/2				
EVOCAÇÃO DIFERIDA		Deve recordar as palavras SEM PISTAS		Boca Linho Igreja Cravo Azul		Pontuação apenas para evocação SEM PISTAS		___/5
Opcional		Pista de categoria		Pista de escolha múltipla				
ORIENTAÇÃO		<input type="checkbox"/> Dia do mês <input type="checkbox"/> Mês <input type="checkbox"/> Ano <input type="checkbox"/> Dia da semana <input type="checkbox"/> Lugar <input type="checkbox"/> Localidade		___/6				
© Z.Nosreddine MD		Examinador: _____		TOTAL		___/30		

Figura 2.2: Questionário MoCA [2]

2.2 Sistemas de Gestão de Inquéritos

Nos dias de hoje existem diversos sistemas de gestão de inquéritos que podem ser utilizados nas mais diversas áreas. Tratam-se maioritariamente de plataformas online que permitem que um utilizador aceda a diversos modelos de questionários, que podem ser partilhados por um grupo de pessoas, para os mais diversos propósitos. De entre os sistemas existentes, destacam-se o Survio, Quidgest, SurveyMonkey e QuestionPro, que serão analisados em seguida.

2.2.1 Survio

O Survio é um sistema de preparação de questionários, recolha e análise de dados e posterior partilha dos mesmos e foi desenvolvido em 2012 [3]. Este sistema foi desenvolvido especialmente para pequenas e médias empresas e para as entidades que têm que preparar os seus próprios questionários, tais como escolas, escritórios e pequenas organizações, entre

outras.

Trata-se de um serviço pago com uma versão gratuita que alberga menos funcionalidades. No entanto, este sistema tem vários modelos de inquéritos prontos a usar e permite exportar os resultados para um relatório em qualquer formato desejado, com tabelas ou gráficos comparativos [3]. Acrescenta-se a facilidade de os questionários modelados serem adaptáveis a várias plataformas digitais, como por exemplo o IOS e o Android.

O principal fator de diferenciação desta plataforma é a compra de respostas. Esta funcionalidade é fornecida por um serviço adicional. O preço total das respostas varia consoante os critérios especificados na configuração e com o tamanho do questionário.

Esta plataforma apresenta, no entanto, uma desvantagem: se o inquérito possuir um elevado número de perguntas, torna-se difícil a navegação.

O Survio, como se pode observar na Figura 2.3, contém diversos modelos de questionário predefinidos que permitem uma escolha a um nível profissional, sendo que estes modelos se dividem em categorias: i) Inquéritos de mercado; ii) Serviços; iii) Feedback do utilizador; iv) Recursos humanos; v) Eventos; vi) Educação; vii) Comunidade e Organizações Não Governamentais (ONGs); viii) Cuidados de saúde; e ix) Outros.

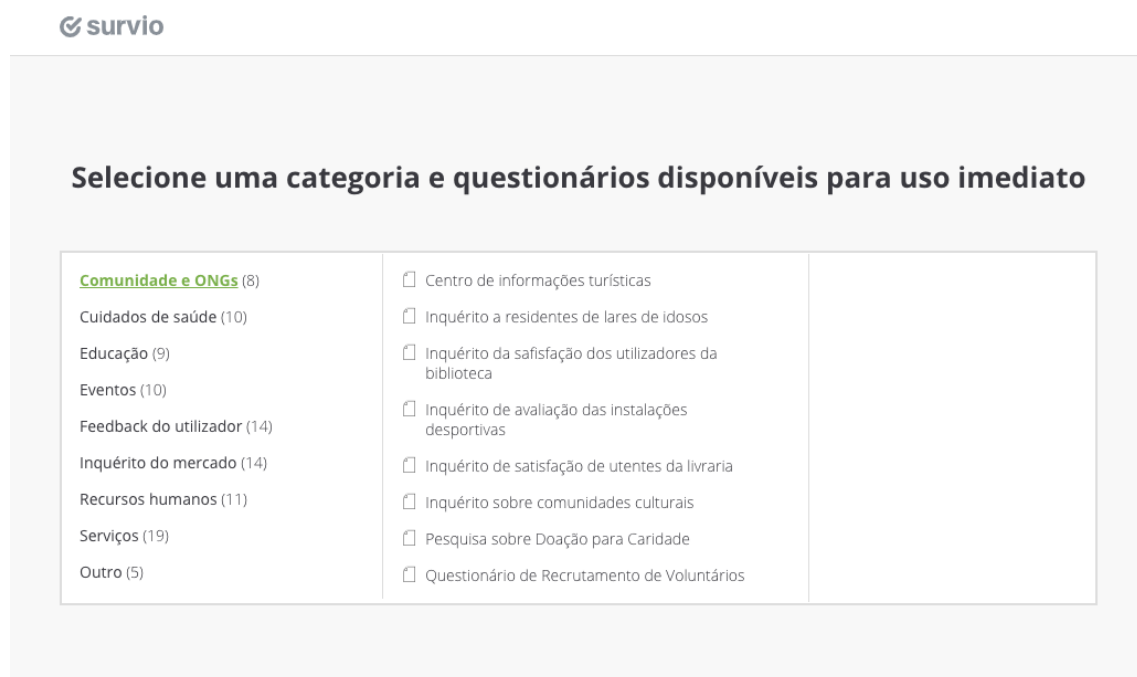


Figura 2.3: Modelos de questionários presentes no Survio [3]

Se nenhum dos modelos já existentes no sistema satisfizer as necessidades do utilizador, também é possível criar um novo questionário que se adeque aos requisitos em mão. Neste caso, o utilizador pode preencher os campos relativos às perguntas do questionário, como se pode ver através da Figura 2.4 e escolher se a resposta é: escolha única, escolha múltipla,

texto, por imagem, classificação, diferencial semântico, escala de classificação, ordem de preferência, matriz de escolha única ou matriz de escolha múltipla. Mediante o tipo de resposta escolhido, as opções de resposta são inseridas e uma nova pergunta é criada. Este processo tem que ser repetido para todas as questões a serem inseridas no inquérito online.

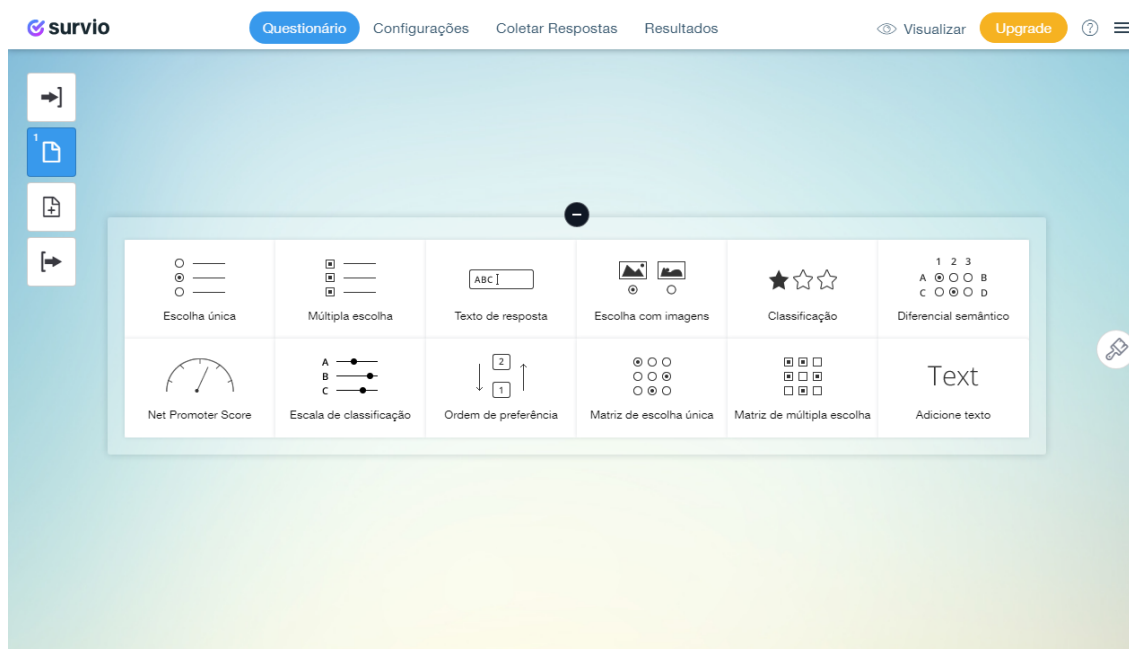


Figura 2.4: Escolha de um tipo de pergunta num questionário feito à medida [3]

2.2.2 SurveyMonkey

A SurveyMonkey é uma plataforma de inquérito online criada em 1999 que atualmente é conhecida por ser uma das principais fornecedoras de serviços de gestão de inquéritos em todo o mundo [4]. Este sistema possui quatro pacotes de funcionalidades, dos quais três são pagos, e serviços que visam auxiliar o criador de cada inquérito. Estes serviços incluem a compra de entrevistados, para serem utilizados no âmbito de entrevistas de emprego, e para a motivação da população geral no sentido de responder aos inquéritos.

Relativamente à interface do sistema, esta acaba por se tornar complexa devido ao número de funcionalidades. Possui muita informação na página de criação dos inquéritos e não existe qualquer guia para o utilizador, o que dificulta a orientação do mesmo.

Tal como o Survio, mencionado anteriormente, os modelos disponibilizados pelo SurveyMonkey, estão inseridos em categorias para ser mais fácil o utilizador saber qual se adequa às suas necessidades ou, caso nenhum seja adequado, se tem de criar um questionário do zero. Os modelos dividem-se em: i) Comunidade, com um modelo; ii) Diversão, com três opções; iii) Educação, com um modelo; iv) Eventos, com cinco opções; v) Feedback de clientes, com seis esquemas disponíveis; vi) Informações demográficas, com uma opção de

questionário; vii) ONG, com cinco opções; viii) Pesquisa de mercado, com dois esquemas; ix) Política, com uma opção predefinida; x) Recursos humanos, com dez tipos de questionários; xi) Saúde, com quatro modelos; e Setoriais, com dez opções.

Apesar de o SurveyMonkey conter modelos de inquéritos relativos à área da saúde, estes só são relativos à avaliação de plano ou seguro de saúde, modelo de dieta e fitness, modelo de avaliação de desempenho hospitalar e modelo de avaliação de desempenho médico.

No caso de nenhum dos modelos predefinidos que a plataforma disponibiliza satisfazer as necessidades do utilizador, então este tem a opção de criar o seu próprio questionário de um modo intuitivo. A criação de um novo questionário inclui a configuração visual do questionário, tal como inserção de um logótipo, por exemplo. É possível importar perguntas de outros questionários já elaborados e criar novas. No caso das novas perguntas, estas são configuradas através de um editor de texto onde se escreve a pergunta e opções de resposta. Estas opções incluem: escolha múltipla, caixas de seleção, avaliação com estrelas, upload de arquivo, caixa de texto única, caixa de comentário, matriz de menus suspensos, lista suspensa, matriz/escala de avaliação, classificação, barra deslizante, várias caixas de texto, informação de contacto e data/hora. É possível também definir se a resposta à nova pergunta do questionário é ou não obrigatória.

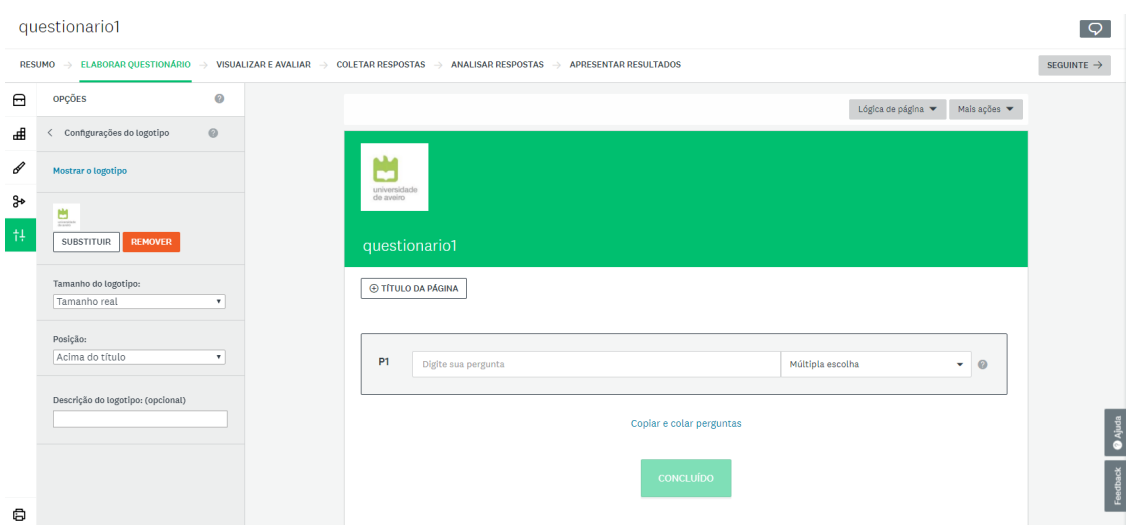


Figura 2.5: Criação de um questionário no SurveyMonkey [4]

Esta plataforma também pede alguns dados específicos ao utilizador na criação da sua conta, dados esses que são utilizados para atribuir algumas configurações ao questionário. Por exemplo, no caso de o email do utilizador pertencer à Universidade de Aveiro, o questionário já será configurado com o respetivo logótipo e numa cor compatível, neste caso o verde do logótipo, como se pode ver na Figura 2.5. Isto é útil porque através de dados já coletados, permite melhorar a apresentação de forma automática e assim, o utilizador tem apenas que adicionar as perguntas e, se assim o desejar, modificar alguma configuração que lhe seja conveniente.

2.2.3 QuestionPro

A QuestionPro é uma plataforma de inquéritos online paga, criada em 2002, cujos principais utilizadores são empresas e organizações sem fins lucrativos, universidades e particulares [5]. Em adição, fornece uma licença gratuita para organizações sem fins lucrativos e disponibiliza um conjunto de recursos que visam a aprendizagem de métodos de elaboração de inquéritos.

Esta plataforma destaca-se das restantes pela capacidade de análise de dados: possui vários métodos de análise de dados como Total Unduplicated Reach and Frequency (TURF), Trend Analysis (descoberta de tendências na informação recolhida) e CrossTables.

A QuestionPro tem um Market de inquéritos, onde o utilizador poderá comprar e vender inquéritos e, analogamente ao SurveyMonkey, possui uma funcionalidade que permite comprar respostas.

No geral, a plataforma é completa e organizada. As configurações do inquérito acompanham todo o processo num menu lateral. Possui um bom método de visualização de dados: para cada pergunta é possível visualizar um gráfico e uma tabela com os respetivos resultados.

Em adição, o sistema também possui aplicações móveis que permitem o acesso aos dados offline qualquer que seja o sistema operativo utilizado, sendo assim fácil e eficaz na acessibilidade dos resultados.

Os vários modelos do QuestionPro estão divididos em questionários com os seguintes propósitos: avaliação do público, pacote básico, cliente e satisfação do cliente, educação, opiniões de funcionários, planeamento de eventos, acompanhamento de incidentes, específicos da indústria, específicos de organizações não governamentais, satisfação de leitores, experiência de compras e serviços e opiniões de produtos no geral.

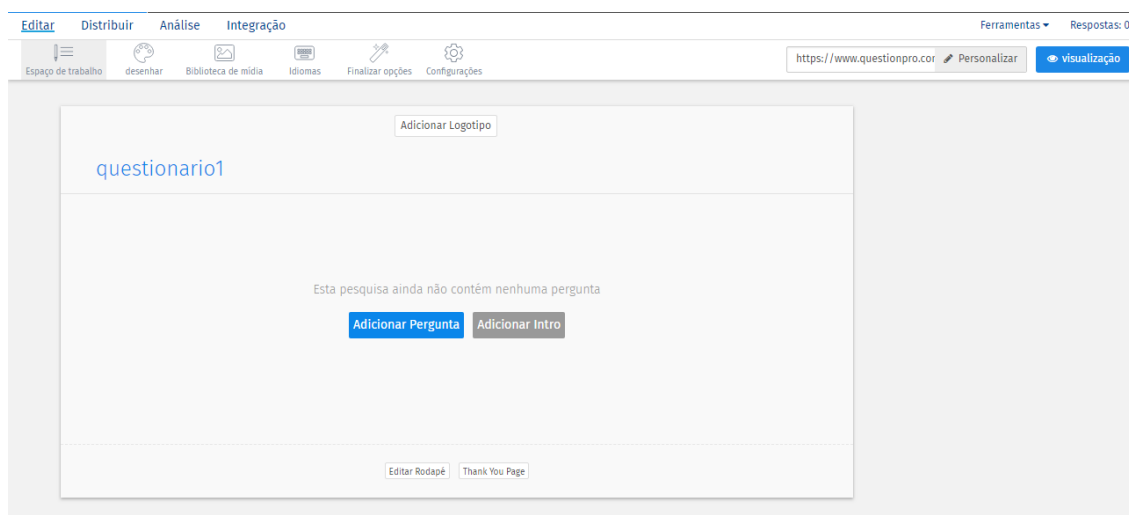


Figura 2.6: Criação de um questionário no QuestionPro [5]

É possível também criar um questionário do zero, como se pode ver na Figura 2.6. O utilizador pode configurar todo o aspeto do questionário, desde adicionar o logótipo até ao uso de temas que permitem definir as cores e fontes. É possível adicionar uma introdução ao questionário e a partir daí, todas as perguntas necessárias. Tal como o SurveyMonkey, este sistema permite a importação de perguntas, já adicionadas noutros questionários, e a criação de novas questões. No caso da adição de uma nova pergunta, é necessário primeiro escolher o tipo de resposta que esta irá ter e, neste ponto, o QuestionPro é muito abrangente, uma vez que apresenta diversas opções que ainda possuem opções dentro delas: i) Escolha múltipla, com opção de selecionar uma, selecionar várias ou menu suspenso; ii) Classificação geográfica, que inclui as opções classificação em estrela, classificação com smile, classificação de polegar para cima/baixo, barra deslizante de texto e slider numérico; iii) Encomenda, que inclui rank order, soma constante ou arrastar e soltar; iv) Artigo estático, que tem as opções apresentação do texto, cabeçalho de secção, secção e sub-rubrica; v) Texto, que tem as opções de caixa de texto, caixa de texto de linha única, endereço de e-mail e informações para contacto; vi) Escolher imagem, dentro da qual há a opção de escolher uma, escolher várias ou classificação; vii) Matriz básica, que inclui a dimensão multi-ponto, seleção de vários elementos e planilha; e viii) Misc, que inclui respostas do tipo data/hora, captcha, calendário e mapas.

Também existem opções de resposta avançadas que contêm ainda mais tipos de respostas possíveis, mas só estão disponíveis na versão paga. Só depois de selecionar o tipo de resposta é que se insere a pergunta e se define se é de resposta obrigatória ou não.

2.2.4 Análise Comparativa

Através de uma análise do funcionamento das plataformas conclui-se que a maioria delas segue um processo de gestão de inquéritos com o seguinte fluxo: definição da formatação do inquérito, construção do inquérito com todas as funcionalidades de gestão de páginas e perguntas, disponibilização do inquérito para o público alvo e visualização dos dados, tanto para exportação como no formato de gráficos e tabelas.

Os sistemas estudados contêm diversas funcionalidades. A Tabela 2.1 pretende comparar alguns dos pontos chave destes sistemas, tal como o facto de permitirem ou não a personalização dos inquéritos, o tratamento e exportação de dados, se permite ou não ter contas de utilizador. Em adição, os sistemas também são comparados a nível de requisitos específicos de um sistema de gestão de inquéritos de apoio à avaliação multidimensional da funcionalidade de pessoas idosas, nomeadamente existência de dashboards de dados, relação gráfica entre inquéritos e reutilização de questões em diferentes inquéritos.

Os sistemas já existentes, para responder à concorrência, vão aumentando as suas funcionalidades. O aumento das funcionalidades proporciona um aumento da complexidade, tornando-se mais difícil criar um inquérito ou questionário.

Em termos de usabilidade, algumas das plataformas analisadas dificultam o acesso às funcionalidades principais: contêm muito conteúdo nas páginas de gestão e criação de inquéritos. Em contrapartida, possuem guias para que o utilizador foque a sua atenção no que pretende.

	Survio	Survey Monkey	Question Pro
Inquéritos Personalizados	✓	✓	✓
Tratamento de dados	✗	✗	✓
Exportação de dados	✓	✓	✓
Contas de utilizador	✓	✓	✓
Armazenamento de dados	✓	✓	✓
Relatórios interativos	✓	✓	✓
Paga	✓	✓	✓
Dashboards dinâmicos	✗	✗	✗
Relação entre os diferentes inquéritos	✗	✗	✗
Reutilização de questões	✗	✓	✓

Tabela 2.1: Tabela de Comparação de Sistemas de Gestão de Inquéritos

A plataforma SurveyMonkey continua a ser das plataformas com mais utilizadores, prima pela qualidade dos serviços e é mais focada no utilizador final. É a plataforma mais competitiva neste mercado porque, para além das vantagens já mencionadas, é das plataformas que pratica preços mais baixos.

Considerando o objetivo da presente dissertação, ou seja, o desenvolvimento de um sistema de gestão de inquéritos de apoio à avaliação multidimensional da funcionalidade de pessoas idosas, os sistemas de inquéritos apresentados não satisfazem os requisitos necessários, uma vez que não são completamente dinâmicos e não permitem o relacionamento entre os dados de diferentes questionários nem a criação de um dashboard geral.

2.3 Base de Dados

Uma Base de Dados (BD) pode ser definida como uma coleção organizada de dados que estão relacionados entre si e que podem ser partilhados por múltiplas aplicações. Para facilitar a definição, construção, manipulação e partilha da BD, usa-se um Sistema de Gestão de Base de Dados (DBMS – Database Management System) [36]. A secção atual pretende apresentar diversas soluções de BDs, tanto relacionais como não relacionais, e executar uma comparação entre estas. Em adição, serão descritas e comparadas várias arquiteturas que podem ser utilizadas no desenvolvimento de software, bem como as web frameworks consideradas e uma explicação e comparação das mesmas.

2.3.1 Relacional vs. Não Relacional

Existem dois tipos principais de BDs: Relacionais e Não relacionais, sendo estes últimos também conhecidos por Not Only SQL (NoSQL). Estes dois tipos diferem na forma de

armazenamento e organização dos dados, sendo que, no caso do modelo relacional a informação é guardada sob a forma de tabelas relacionadas entre si.

As tabelas são formadas por diversas colunas, cada uma com o seu tipo de dados, onde estes serão posteriormente guardados. Toda a definição, manipulação e recuperação de informação em BDs relacionais é feita através de queries Structured Query Language (SQL), enquanto que no caso de uma BD não relacional, o mesmo processo é feito usando uma Application Programming Interface (API) orientada a objetos [37].

2.3.2 Base de Dados Relacional

Para trabalhar com uma BD relacional, no caso específico do SQL, pode-se usar diversas ferramentas, tal como o SQL Server, que é responsável pelo armazenamento e recuperação de dados requisitados por outras aplicações de software. Existe também o SQL Server Management Studio, uma aplicação de software que foi lançada com o SQL Server, que é utilizada para configurar, gerir e administrar todos os componentes do SQL Server [38].

A nível esquemático, o modelo relacional, apesar de simples, é muito pouco flexível, uma vez que os dados são sempre armazenados em tabelas normalizadas. No entanto, o facto de a informação estar guardada num formato predefinido torna o acesso aos dados muito mais eficiente. Em adição, a linguagem SQL consiste em três tipos de queries, sendo estas: Data Definition Language (DDL), Data Control Language (DCL) e Data Manipulation Language (DML). Cada uma destas tem um conjunto de queries que são executadas para definir e manipular dados e controlar transações [39].

A nível de BDs Relacionais, os seguintes sistemas de gestão de BD serão analisados: MySQL, Oracle e SQL Server.

MySQL

MySQL é um Sistema de Gestão de BD relacional de domínio público distribuído pela Oracle. Como se trata de uma BD do modelo lógico relacional, é constituída por objetos, os quais podem ser outras BDs, tabelas, visualizações, linhas e colunas. É possível criar relações entre os diferentes campos de dados, sejam estas de um para um, um para muitos ou vice-versa, únicas, obrigatórias ou opcionais, criando assim ponteiros entre tabelas [40]. A parte SQL do nome significa “Structured Query Language”, o que implica que é esta a linguagem utilizada para aceder às BDs. Em adição, o MySQL apresenta características como a portabilidade, uma vez que escrito em C e C++, testado com diferentes compiladores e obedece a um design de servidor com várias camadas com módulos independentes; diferentes tipos de dados; declarações e funções; segurança (i.e. possui segurança por passwords encriptadas para todo o tráfico correspondente às mesmas na conexão com o servidor e que usa um sistema de passwords e privilégios); escalabilidade, (i.e. oferece suporte para BDs de dimensões grandes; conectividade, através de diversos protocolos como, por exemplo, sockets TCP/IP; localização, através de disponibilização de diversas mensagens de erro; e por fim aplicações cliente e ferramentas, na medida em que o MySQL inclui diversas aplicações de suporte [41].

Oracle

O Oracle é um DBMS multi-modelo criado e distribuído pela Oracle Corporation. Trata-se de um modelo relacional que utiliza linguagem SQL e que garante a independência entre o armazenamento físico dos dados e as suas estruturas lógicas. Nesta BD, especificamente, um schema é uma coleção de estruturas de dados lógico e contém objetos, que são estruturas criadas pelo utilizador para referenciar os dados armazenados. Os dois principais objetos schema suportados são tabelas que, tal como no MySQL, servem para descrever uma entidade e contém várias colunas, cada uma com um nome, tipo e dimensão para os dados, índices e estruturas opcionais que podem ser criadas numa ou mais colunas da tabela para melhorar a recuperação de informação [6].

O servidor de uma BD Oracle consiste numa BD e pelo menos uma instância da mesma, que consiste num conjunto de ficheiros, localizados no disco, que armazena os dados, e um conjunto de processos em segundo plano que gerem os ficheiros da BD. Este último consiste numa área de memória partilhada, System Global Area (SGA). A Figura 2.7 apresenta a BD e a sua instância, tendo em conta que um processo de cliente tem que ser executado cada vez que o utilizador se conecta à instância. Cada processo de cliente está associado a um processo de servidor com a sua própria memória de sessão, Program Global Area (PGA) [6].

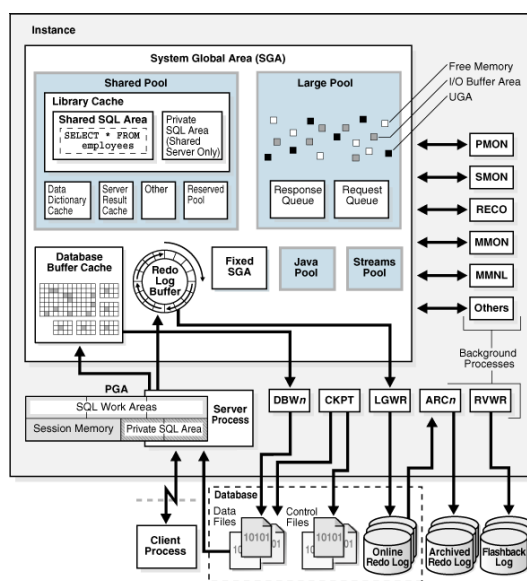


Figura 2.7: Instância e Base de Dados Oracle [6]

SQL Server

O Microsoft SQL Server é um DBMS relacional desenvolvido pela Microsoft. Tem como principal objetivo o armazenamento e recuperação de informação como é requisitado por

outras aplicações.

O SQL Server inclui diversas componentes de servidor que incluem diferentes ferramentas e servem diversos propósitos, tal como o SQL Server Database Engine que inclui o motor de BD, os serviços para armazenar, processar, proteger e replicar dados, mecanismos de pesquisa no texto completo, ferramentas para gerir dados relacionais e Extensible Markup Language (XML), integração de análise da Base de Dados, integração polybase para acesso a outras fontes de dados heterogéneas e servidor de Data Quality Services (DQS) [42].

Para gestão do SQL Server, existem diversas ferramentas, no entanto, o mais utilizado e também mais completo, é o SQL Server Management Studio, um ambiente integrado para aceder, configurar, gerir, administrar e desenvolver componentes do SQL Server [42].

2.3.3 Base de Dados Não Relacional

Existem quatro tipos de BDs NoSQL, cada uma com as suas características: BDs com estrutura em formato de gráfico, na qual os dados são conectados entre si em formato de gráfico; Armazenamento chave-valor; na qual o armazenamento é feito através de uma chave e o valor a ela associado; Armazenamento em colunas, onde os dados são organizados de modo a formar secções de colunas de dados; e por fim, BD baseada em documentos, que é uma versão melhorada do Armazenamento chave-valor com a adição de os pares serem guardados como documentos que representam estruturas complexas, como por exemplo, a JavaScript Object Notation (JSON) [39].

Um modelo de BD não relacional traz vantagens a nível de eficiência quando se trata de dados não uniformes, isto é, na gestão de conjuntos de dados de grande dimensão que sofrem modificações constantes, uma vez que não seguem esquemas restritos [39].

A MongoDB é um exemplo de uma BD NoSQL que representa os dados em coleções de documentos JSON com esquema e organização dinâmicos, ou seja, é constituída por um conjunto de coleções, sendo que estas não têm um formato predefinido como acontece no caso das BDs relacionais. Neste caso, são utilizados pares de chave-valor, nos quais a chave funciona como um identificador único de determinados dados e o valor serão os dados propriamente ditos. Isto implica que todos os documentos permitem leitura/escrita com a utilização da chave [43].

MongoDB armazena os registos dos dados em documentos BSON (representação binária de JSON), que contém mais tipos de dados do que o JSON. Para além de guardar os dados, estes documentos são utilizados para, remotamente, fazer pedidos à BD [44].

2.3.4 Comparação entre DBMS

Enquanto que o MongoDB é uma BD NoSQL que se baseia no armazenamento e manipulação de dados em documentos JSON, o MySQL, o Oracle e o SQL Server são relacionais e usam tabelas que referenciam dados umas das outras de modo a criar uma base de dados coesa.

Como já foi mencionado anteriormente, um modelo de BD não relacional tem vantagens a nível de eficiência na gestão de conjuntos de dados de grande dimensão que sofrem modi-

ficações constantes, uma vez que não seguem esquemas restritos.

Por outro lado, a escolha de uma BD relacional é indicada para aplicações que envolvem a gestão de diversas transações e com consultas e análise de dados complexa, uma vez que é necessário garantir que as transações sejam seguras e eficazes. A estrutura de uma BD do modelo relacional permite a ligação entre informação de diferentes tabelas através do uso de chaves estrangeiras ou de índices que identificam determinados dados dentro da BD. As outras tabelas terão uma referência para essa chave estrangeira, de modo a criar uma relação clara entre os dados [37].

Pelos motivos citados anteriormente, o modelo relacional será o mais indicado tendo em mente uma aplicação na qual existe uma grande complexidade a nível de queries, transações e análise de dados [37]. É também uma solução mais simples tendo em conta que a informação a ser armazenada e manipulada tem sempre o mesmo formato.

2.4 Arquiteturas

Nesta secção serão apresentados os diferentes padrões arquiteturais que descrevem uma abordagem para o desenvolvimento de software, bem como as suas vantagens e desvantagens e uma comparação entre estes.

2.4.1 Arquitetura Model-View-Controller

O uso da arquitetura Model-View-Controller (MVC) permite separar a aplicação em três componentes principais interligadas entre si: Modelo, Visualização e Controlador. Nesta arquitetura, presente na Figura 2.8, os pedidos do utilizador são encaminhados para o Controlador, que é responsável pela realização das diversas ações e o acesso aos dados da BD através de queries. O Controlador seleciona a Visualização a ser apresentada ao utilizador e disponibiliza os dados que este possa precisar. Desta maneira, tanto a Visualização como o Controlador dependem do Modelo, no entanto, como o Modelo não depende de nenhum dos outros componentes, esta separação permite que o Modelo seja compilado e testado independentemente da apresentação visual que foi escolhida [45].

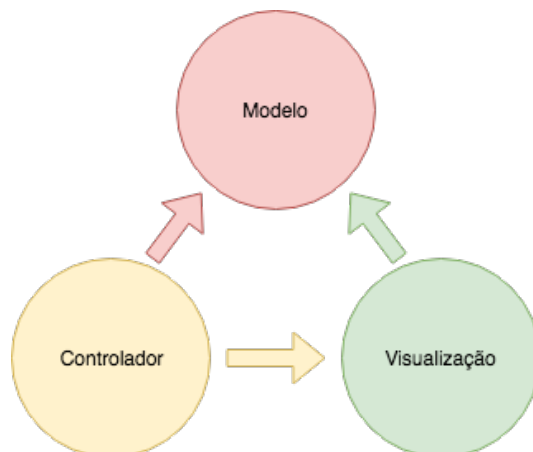


Figura 2.8: Diagrama de relação entre os componentes do MVC

O MVC apresenta grandes vantagens como, por exemplo, o facto de não utilizar estados de visualização ou formulários baseados no servidor ou suportar o desenvolvimento orientado a testes [46].

Existe uma variante do MVC, designada por MTV, que usa Templates em vez de Controladores.

2.4.2 Arquitetura Model-View-Presenter

A arquitetura Model-View-Presenter(MVP) é derivada do MVC, utilizando um Presenter que assume todas as funcionalidades do Controlador na outra arquitetura. Nesta arquitetura em concreto, a Visualização é responsável pela manipulação dos eventos da Interface de Utilizador, enquanto que o Modelo funciona como um modelo de domínio. Apesar de apresentar grandes semelhanças com o MVC, o MVP é mais focado na interface com o utilizador [7].

Existem dois tipos de implementação: Passive View, cujo funcionamento é ilustrado na Figura 2.9, e Supervising Presenter, presente na Figura 2.10. No caso do primeiro tipo de implementação, existe supervisão do Presenter, na medida em que a Visualização não é responsável pela sua própria atualização com base no Modelo, dado que isso fica a encargo do Presenter, que faz a ligação com os outros componentes. No caso do Supervising Controller, o Presenter tem como função a implementação da lógica de visualização e a comunicação entre a Visualização e o Modelo é feita através de data binding, isto é, através de uma técnica geral que une duas fontes de dados e as mantém sincronizadas entre si [7]. A organização do MVP simplifica a utilização de testes unitários, uma vez que a Visualização só acede ao Presenter através de uma interface que pode ser facilmente simulada para efeitos de teste [46].

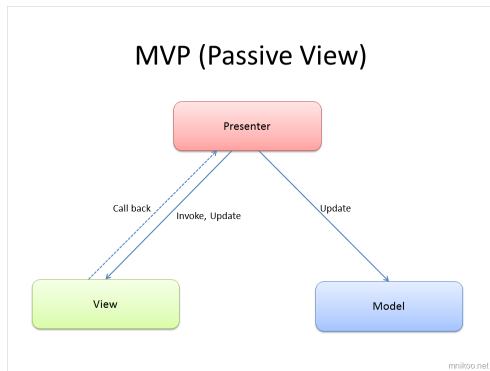


Figura 2.9: MVP Passive View [7]

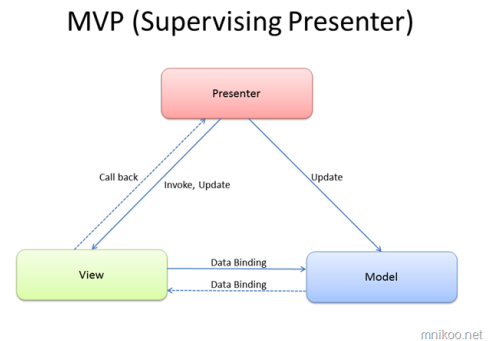


Figura 2.10: MVP Supervising Presenter [7]

2.4.3 Arquitetura Model-View-ViewModel

A arquitetura Model-View-ViewModel (MVVM) é uma especialização do MVP conhecida como Modelo de Apresentação que foi criada tendo em mente os ambientes Windows Presentation Foundation (WPF) e Silverlight. É constituída por três componentes, tal como as arquiteturas apresentadas anteriormente e tem semelhanças com o MVC, na medida em que duas das componentes, o Modelo e a Visualização, funcionam da mesma maneira. Em vez de Controlador ou Presenter, o terceiro componente é o View Model, como se pode ver na Figura 2.11, o modelo da visualização que estende as propriedades do Modelo que são utilizados na Visualização e que desempenha as funções que pertenceriam ao Controlador numa arquitetura MVC. Em adição, tem que existir suporte para ligação entre os dados presentes na Visualização e no Modelo, o que acaba por ser uma grande desvantagem, uma vez que antes do ambiente WPF, não era possível implementar uma aplicação complexa por causa da falta de suporte para data binding e, mesmo com WPF, não é trivial desenvolver uma aplicação sem suporte adicional [7].

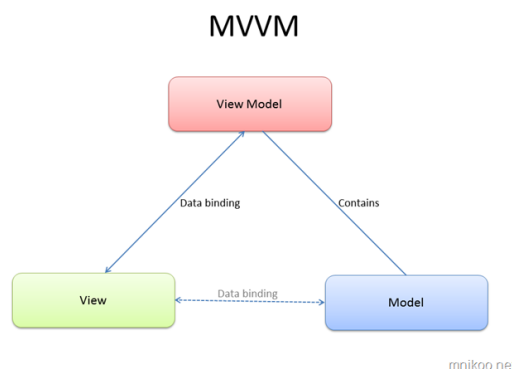


Figura 2.11: Diagrama de relação entre os componentes do MVMM [7]

2.4.4 Comparação entre Arquiteturas

Apesar de as três arquiteturas apresentadas conterem semelhanças, as suas características acabam por as diferenciar com vista a um desenvolvimento adequado para uma dada aplicação.

O MVP é aconselhado para situações onde não é possível fazer a ligação dos dados através do seu contexto, como o caso específico do Windows Forms, que separa a Visualização do Modelo através do Presenter e no qual a Visualização não pode ser conectada diretamente com o Presenter, transmitindo informação através de uma interface [46]. Em adição, também permite testes unitários.

Por outro lado, o MVMM é, normalmente, usado quando é possível fazer a ligação de informação através do seu contexto [46].

Por fim, o MVC é utilizado em situações onde a conexão entre a Visualização e o resto do programa não se encontra sempre disponível, eliminando assim as hipóteses de usar o MVP e o MVMM de modo eficiente. Um bom exemplo de framework que permite a gestão destas situações é a ASP.NET MVC da Microsoft [46].

2.5 Frameworks Web

As frameworks web escolhidas para comparação são ASP.NET e Django. As próximas secções irão centrar-se nas características de ambas as frameworks e no ambiente de desenvolvimento utilizado para as mesmas.

2.5.1 ASP.NET Model View Controller

ASP.NET é uma plataforma da Microsoft que permite o desenvolvimento de aplicações Web através da criação de páginas dinâmicas numa linguagem de programação integrada na framework .NET. Todas as aplicações desenvolvidas para esta plataforma são escritas em C Sharp ou Visual Basic e são executadas sobre um servidor Internet Information Services (ISS).

C Sharp é uma linguagem de programação multi-paradigma desenvolvida pela Microsoft como parte da plataforma .NET, projetada para funcionar na Common Language Infrastructure da framework e que se baseia noutras linguagens como C, C++ e Java. Esta linguagem de programação visual é orientada a objetos e permite que componentes de software de diferentes linguagens possam interagir entre si.

As aplicações nesta linguagem podem usar protocolos como o Simple Object Access Protocol (SOAP), que é um protocolo de troca de mensagens com informação estruturada na implementação de serviços web, e linguagens como o Extensible Markup Language (XML) para interagir com a internet.

As principais características do C Sharp são: simplicidade; flexibilidade; o facto de ser orientada a objetos, que implica que as variáveis têm que fazer parte de classes; o facto de usar variáveis com tipos específicos, o que ajuda a evitar erros por manipulação imprópria de tipos e atribuições incorretas; ser capaz de gerar código automático; possuir controlo de

versões, uma vez que cada assembly gerado, em formato DLL ou EXE, contém informação sobre a versão do código; suportar a código legado de objetos COM e DDL; e ainda o facto de a gestão de memória ser feita durante o runtime, existindo para tal uma entidade designada por Garbage Collector.

O ASP.NET tem três tipos de frameworks: Web Forms, MVC e Web Pages. A escolha entre estas três depende apenas dos requisitos de uma determinada aplicação, no entanto, só o ASP.NET MVC é que é constituído por uma arquitetura MVC [47].

Como foi mencionado anteriormente, o ASP.NET MVC separa a aplicação em três componentes principais interligadas entre si: Modelo, Visualização e Controlador.

- Numa aplicação MVC, o Modelo representa o estado da aplicação e o comportamento desta que não seja específico da interface de utilizador, bem como a implementação da lógica e conceptual associada ao domínio de dados. Pode fazer sentido um aplicação complexa ter diferentes tipos de modelos com diferentes responsabilidades: Modelo de domínio (Domain Model), que inclui abstrações e serviços que permitem ao controlador operar num nível superior de abstracção; Modelo de Visualização (View Model), que é usado especificamente para conter os dados que uma certa visualização apresenta e, em alguns casos, retorna para o servidor, contendo também qualquer lógica de tratamento dos dados para posterior apresentação dos mesmos; Modelo de Ligação (Binding Model), que só contém dados e nenhuma lógica comportamental, usados para evitar problemas de segurança relativos à ligação do modelo, impedindo assim que utilizadores acessem a propriedades do modelo que não estão presentes no formulário; Modelo API (API Model), no caso de a aplicação usar uma API e o formato dos dados apresentados aos clientes ser separado do modelo de domínio interno; e Modelo de Persistência (Persistence Model), que consiste em classes usualmente geradas com base no esquema da base de dados e que mapeia como é que os dados são recuperados e armazenados na BD.
- A Visualização apresenta o conteúdo através de uma interface de utilizador, fazendo uso de código C Sharp embutido em HyperText Markup Language (HTML). Assim, estamos perante uma linguagem de Templates que usa código C Sharp embutido e que gera dinamicamente conteúdo web no lado do servidor. A lógica utilizada nas Visualizações é mínima e relacionada com a apresentação dos dados.
- No caso do Controlador, este trata da interação com o utilizador e com o modelo e, em última instância, selecciona a visualização a ser apresentada na interface do utilizador. No modelo MVC, os Controladores são os pontos de entrada responsáveis por seleccionar os tipos de Modelos para obter, criar os dados e as Visualizações a ser mostradas e por tratar e responder às inserções de dados por parte do utilizador.

ASP.NET Core MVC inclui as seguintes características: encaminhamento; ligação do modelo; validação do modelo; injeção de dependências; filtros; áreas; Web APIs; testabilidade; Razor view engine; Visualizações; Tag Helpers e componentes de visualização.

O fundamento deste modelo é o encaminhamento de ASP.NET Core, um componente

de mapeamento de URLs que permite a definição dos padrões de nomeação dos Uniform Resource Locators (URLs) de modo a que estes funcionem corretamente em termos de otimização de motores de pesquisa e de geração de apontadores, independentes de como os ficheiros estão organizados no servidor web.

O encaminhamento pode ser definido usando uma sintaxe de encaminhamento que suporta restrições de valores, bem como valores predefinidos e opcionais. O encaminhamento baseado em convenções permite definir globalmente os formatos de URL aceites e como é que cada formato em específico mapeia uma ação num dado controlador. Neste caso, quando um pedido é recebido, o motor de encaminhamento analisa o URL e compara-o com os formatos de URL aceites e chama o método do controlador associado. Por outro lado, o encaminhamento por atributos permite especificar a informação de encaminhamento através de atributos quer nos controladores, quer nas ações, que definem as rotas da aplicação.

A ligação do modelo converte os dados pedidos pelo cliente em objetos sob os quais o controlador pode operar. Assim, a lógica no controlador não tem que tratar os dados que recebe, apenas os recebe como parâmetros nos seus métodos.

MVC suporta a validação através de atributos de validação nos objetos modelo chamados de Data Annotation. Isto permite que os diferentes atributos sejam verificados antes de serem enviados para o servidor.

Em adição, existe suporte integrado para injeção de dependência (dependency injection – DI), na medida em que os controladores podem solicitar serviços através dos seus construtores, seguindo o princípio de dependências explícitas, que afirma que métodos e classes devem explicitamente solicitar quaisquer objetos relacionados necessários ao seu funcionamento.

Os filtros permitem a execução de código antes ou depois de fases específicas do pipeline que processa os pedidos. Podem ser aplicados aos controladores ou ações como atributos ou executados globalmente, como por exemplo para tratar exceções ou definir autorizações.

Para particionar uma aplicação deste género em grupos funcionais, criam-se estruturas dentro da aplicação chamada áreas. Os componentes lógicos como Modelo, Controlador e Visualização ficam localizados em diferentes pastas e os seus nomes são usados para estabelecer uma relação entre os componentes.

Para além de permitir a construção de sites web, ASP.NET MVC suporta a construção de WEB APIs. A framework inclui suporte para formatar dados em JSON ou XML ou até mesmo para formatos próprios.

Fatores mencionados anteriormente, como o uso de interfaces e a injeção de dependências, tornam o MVC apto para teste unitário. A framework também inclui certas características que permitem testes de integração fáceis e rápidos.

Quanto ao motor de Visualização Razor, utilizado para fornecer e definir as Visualizações com código C Sharp embutido, também permite gerar dinamicamente conteúdo web do lado do servidor, permitindo assim misturar código proveniente do servidor com o código e conteúdo do lado do cliente. Usando este recurso, é possível definir esquemas de página, visualizações parciais e secções substituíveis. Em adição, as Visualizações podem ser construídas com base no Modelo correspondente, uma vez que os Controladores podem passar os dados presentes no modelo para as visualizações, o que resulta no facto de estas últimas

possuam verificações de tipos e suportem IntelliSense, um recurso chave que gera código automaticamente no editor de código. Além disto, existem componentes de visualização que proporcionam lógica e reutilizam-na em diversas partes da aplicação. São semelhantes às visualizações parciais com a adição da lógica associada.

Por fim, os Tag Helpers permitem que seja possível criar e mostrar elementos HTML nos ficheiros Razor através da definição de identificadores personalizados ou da modificação de identificadores já existentes. Os Tag Helpers são vinculados para elementos específicos com base nos nomes dos elementos e seus atributos e fornecem benefícios quanto à apresentação de informação do servidor preservando a edição de HTML. Existem muitos identificadores incorporados que são usados para tarefas básicas e comuns, tal como a criação de links e formulários, e muitos outros disponíveis nos pacotes NuGet. Estes são autorizados em C Sharp e têm como objetivo permitir aceder aos atributos que se encontram do lado do servidor através de código executado no lado do cliente. [45]

2.5.2 Django

O Django é uma framework web que usa Python como linguagem e que possui uma arquitetura Model-Template-View (MTV), como se pode ver na Figura 2.3, que se baseia no modelo MVC. Duas das camadas desta arquitetura são Modelo e Visualização, tal como no modelo MVC, no entanto os Controladores são substituídos por Templates. No caso desta framework, o Modelo é uma camada de abstração para manipulação de dados da aplicação web, a Visualização contém toda a lógica responsável por processar os pedidos de um utilizador e responder adequadamente e o Template é responsável por processar a informação que é apresentada ao utilizador [48].

A diferença entre o modelo de arquitetura do ASP.NET MVC e o Django é a existência de Templates, que consiste em partes de HTML com a saída desejada e outras partes com uma sintaxe específica que descreve como é que o conteúdo dinâmico será inserido na página final que é apresentada ao utilizador.

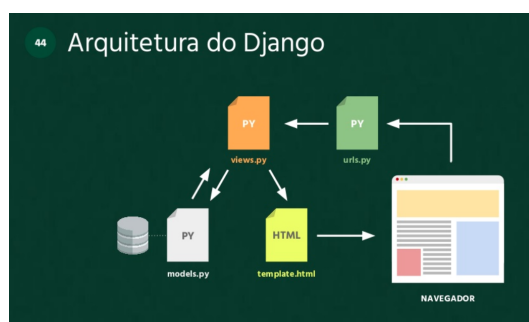


Figura 2.12: Arquitetura MTV [8]

2.5.3 Ambiente de Desenvolvimento

O Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software e web, i.e., desenvolver programas de computador, sites web, serviços web e aplicações móveis [49].

Em termos práticos, o Visual Studio é utilizado para ver e editar código e, posteriormente, desenvolver, fazer debug e publicar uma aplicação. Para isto, usa recursos como validação de arquitetura dinâmica, navegação do código, IntelliSense, refatoração e correções de código que tornam o processo de desenvolvimento mais eficiente. Em adição, suporta 36 linguagens de programação diferentes, incluindo linguagens nativas como o C, C++, Command-Line Interface (CLI), Visual Basic .NET, C Sharp; F Sharp; Javascript, TypeScript, XML, Extensible Stylesheet Language Transformations (XSLT), HTML e Cascading Style Sheets (CSS). Através de plugins adicionais, é possível obter suporte de maneira a utilizar outras linguagens como, por exemplo, o Python, Ruby, Node.js e M [49].

Capítulo 3

Arquitetura

O capítulo atual visa apresentar o design funcional da solução proposta de modo a cumprir os objetivos propostos. Assim, será apresentada a modulação do problema através de casos de uso, requisitos do sistema, sejam estes funcionais ou não funcionais, e mockups de interface com o utilizador.

3.1 Contexto

A aplicação a ser desenvolvida no contexto desta dissertação será uma plataforma web que apoie a avaliação multidimensional da funcionalidade de pessoas idosas. Para este efeito, o sistema terá que suportar diversos instrumentos vocacionados ou globais com as suas questões e pontuações, caso seja caso disso, bem como as respostas permitidas para cada pergunta. O sistema deve possibilitar a seleção dos instrumentos mais adequados à avaliação de um determinado utente, bem como registar todas as avaliações a ele feitas. Em adição, o sistema deverá providenciar mecanismos de acesso a todos os registos efetuados e mostrá-los de forma lógica na página desse utente, permitindo uma análise e perceção rápida por parte do prestador de cuidados.

3.2 Requisitos Funcionais

Os requisitos funcionais são as competências necessárias que o sistema tem que exibir para a resolução do problema em questão. Estes requisitos podem ser separados em diversos subgrupos [50]: requisitos de dados, que definem como é que a aplicação armazena e gere a informação; requisitos de interface, que são relacionados com a interação do utilizador com a aplicação; requisitos de navegação, que representa a navegação do utilizador através do sistema; requisitos personalizados, necessários para que a aplicação se adapte às características dos utilizadores finais e requisitos transacionais, que descrevem as operações internas que são independentes da interface ou interação com o utilizador. Tendo em conta os subgrupos de requisitos funcionais citados anteriormente e o sistema em causa, é possível definir os seguintes requisitos:

- **O sistema deve permitir que o utilizador inicie sessão como administrador ou utilizador comum.** Enquanto que o administrador tem acesso a todo o sistema, podendo assim gerir os utilizadores e seus papéis no sistema, bem como todos os elementos relativos às avaliações da funcionalidade de pessoas idosas, o utilizador comum pode apenas aceder e guardar dados de utentes e dos instrumentos utilizados na avaliação destes.
- **O sistema deve permitir que o utilizador terminar a sua sessão no sistema.** Depois de iniciar sessão e executar as operações pretendidas, é possível terminar sessão.
- **O sistema deve permitir que novos utilizadores se registem no sistema.** Dado um novo utilizador na plataforma, o registo permite a criação de um novo utilizador mediante o preenchimento dos seus dados pessoais num formulário específico.
- **O sistema deve permitir que um utilizador modifique os seus dados pessoais e de sessão.** Tanto o utilizador comum como o administrador podem modificar os seus próprios dados, tal como nome, data de nascimento, morada, email e palavra passe.
- **O sistema deve permitir que o utilizador veja a lista de utentes já inseridos no sistema e os respetivos detalhes.** Um utilizador que já tenha um papel definido pode aceder à lista de utentes, pesquisar por um em concreto e ver os detalhes desse utente, bem como todos os instrumentos a ele aplicados e seus resultados ao longo do tempo.
- **O sistema deve permitir que o utilizador adicione um novo utente.** Um utilizador que já tenha um papel definido pode adicionar novos utentes através do preenchimento de um formulário que inclui os dados pessoais do utente.
- **O sistema deve permitir eliminar um utente.** Um utente nunca é completamente eliminado do sistema, no entanto, se o prestador de cuidados quiser eliminar o utente, os dados deste e as avaliações a ele executadas ficarão como não visíveis na base de dados e não serão apresentadas na plataforma.
- **O sistema deve permitir ao utilizador adicionar uma nova avaliação.** Depois de iniciar sessão, dado um utilizador com um papel atribuído pelo administrador, é possível adicionar uma nova avaliação. Para tal, é necessário escolher o utente através do nome, a data em que a avaliação foi realizada, o instrumento a ser utilizado através do nome do mesmo e a instituição em que a avaliação foi realizada.
- **O sistema deve permitir a edição das avaliações por parte do utilizador.** Dado um utilizador com sessão iniciada e com um papel atribuído, este pode editar todos os dados relativos às avaliações submetidas exceto o instrumento utilizado.

- **O sistema deve permitir ao utilizador ver a lista de instituições disponíveis no sistema.** Um utilizador com um determinado papel e com sessão iniciada pode ver a lista das instituições presentes no sistema e seus detalhes.
- **O sistema deve permitir ao utilizador ver os dados e localização de uma determinada instituição.** Dado um utilizador com sessão iniciada e com um papel atribuído, este pode aceder aos dados de uma instituição presente no sistema e a sua localização no mapa.
- **O sistema deve permitir submeter respostas às perguntas de determinado instrumento.** Depois de criar uma entrada e escolher os seus parâmetros, o utilizador pode preencher os campos das respostas às perguntas do instrumento previamente escolhido e submetê-las ao sistema.
- **O sistema deve permitir ver os resultados de avaliações prévias de um determinado utente.** Dado um utente, o utilizador com sessão iniciada e com um papel previamente atribuído pelo administrador tem acesso a gráficos com os resultados do utente nos diferentes instrumentos usados na sua avaliação ao longo do tempo.
- **O sistema deve permitir que o administrador adicione novos utilizadores.** Os utilizadores criados pelo administrador são autorizados a utilizar o sistema na medida em que este lhes atribui um papel na sua criação.
- **O sistema deve permitir que o administrador remova utilizadores.** Só o administrador pode remover utilizadores.
- **O sistema deve permitir que o administrador modifique o papel de um utilizador.** De modo a garantir que o sistema só é usado por utilizadores autenticados, o administrador é o único tipo de utilizador que pode atribuir papéis aos utilizadores. Qualquer pessoa se pode registar no sistema, no entanto, só passam a ter acesso a este quando lhes é atribuído um papel pelo administrador do mesmo.
- **O sistema deve permitir que o utilizador adicione, edite e elimine instrumentos, questões, valores, tipos de respostas e categorias.** Ao adicionar um instrumento, o utilizador pode definir as respetivas questões, valores, tipos de resposta e categorias. O instrumento pode ser eliminado e as suas respostas editadas.
- **O sistema deve permitir que o utilizador adicione, edite e elimine instituições.**

3.3 Requisitos não Funcionais

Ao contrário dos requisitos funcionais apresentados na secção anterior, os requisitos não funcionais não estão relacionados com a funcionalidade do sistema, mas sim com critérios

que visam avaliar a performance geral do mesmo. Exemplos de requisitos não funcionais são requisitos de usabilidade, requisitos de desempenho, requisitos de portabilidade ou requisitos de segurança.

No contexto desta dissertação, é possível distinguir os seguintes requisitos não funcionais:

- **O acesso aos dados através da aplicação deve ser quase imediato.** A informação a ser apresentada ao utilizador deve ser aparecer na página requisitada no espaço de alguns segundos, de modo a garantir que a aplicação garante um tempo de resposta mínimo.
- **A aplicação deve ser intuitiva e fácil de utilizar.** Navegar nas diversas páginas da aplicação deve ser um processo fácil e intuitivo, bem como executar uma determinada operação no sistema, de modo a garantir uma boa experiência por parte do utilizador.
- **A aplicação deve permitir que vários utilizadores acedam aos mesmos dados ao mesmo tempo.** Os utilizadores podem, em acessos concorrentes, aceder à mesma informação. O administrador pode ter acesso aos dados bem como alterá-los.
- **A aplicação deve ser escalável.** Isto implica permitir novas integrações no sistema, tal como novos instrumentos e testes que sejam necessários no futuro.
- **A aplicação deve ser multi-plataforma.** Um utilizador pode aceder e navegar no sistema através de computador ou tablet.
- **A aplicação deve garantir segurança dos dados.** O sistema deve estar conforme com a legislação e regulamento aplicáveis à proteção de dados pessoais, incluindo a diretiva comunitária 95/46/CE. Assim, todos os dados relativos aos utilizadores, utentes e avaliações, bem como todos os restantes dados guardados no sistema só devem ser acedidos por utilizados com credenciais adequadas e com papéis que permitam o acesso.

3.4 Perfis de Utilização

No sentido de cumprir os requisitos apresentados anteriormente, existem dois atores no sistema:

- **Prestador de cuidados**
Este ator representa um prestador de cuidados formal como, por exemplo, o médico de família, médico hospitalar, fisioterapeuta, enfermeiro, psicólogo ou cuidador de âmbito social. Para além desta distinção entre os prestadores de cuidados, alguns poderão assumir o papel de gestor de caso.
O prestador de cuidados deve ter acesso ao sistema através da sua conta de utilizador,

podendo assim adicionar, elencar, editar e remover utentes, bem como adicionar novas avaliações, elencá-las, editar os seus detalhes e removê-las, caso sejam da sua autoria. Qualquer pessoa pode-se registar no sistema, mas só se tornam utilizadores oficiais do sistema quando o administrador atribuir um papel a essa conta. Os prestadores de cuidados podem sê-lo a título individual ou estar associados a instituições.

- Administrador

O administrador do sistema deverá ser responsável pela gestão e atribuição de papéis aos utilizadores. Isto incluirá criar, editar e apagar contas de utilizadores.

Em adição, também será responsável pela gestão dos instrumentos presentes no sistema para cada categoria, bem como as questões e respetivos tipos de resposta, e pela gestão de instituições.

3.5 Casos de Uso

De modo a aprimorar a compreensão do sistema, as Figuras 3.1 e 3.2 apresentam as principais funcionalidades do sistema consoante os diferentes atores e estas serão também descritas em pormenor.

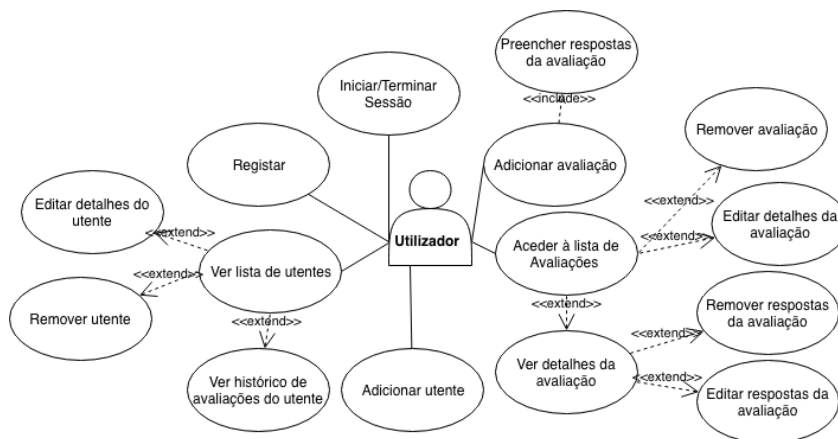


Figura 3.1: Diagrama de casos de uso do Utilizador comum

A aplicação deverá ser criada tendo como utilizador alvo os prestadores de cuidados. Estes serão considerados os utilizadores comuns do sistema e um dos atores. Os prestadores de cuidados terão acesso aos dados da aplicação e que podem preencher, editar e remover utentes e avaliações. Existirão também um ou mais administradores que serão responsáveis

por executar todas as funções administrativas do sistema, tal como gestão dos utilizadores, o que incluirá a atribuição de papéis aos utilizadores comuns, e dos instrumentos de avaliação e seu funcionamento.

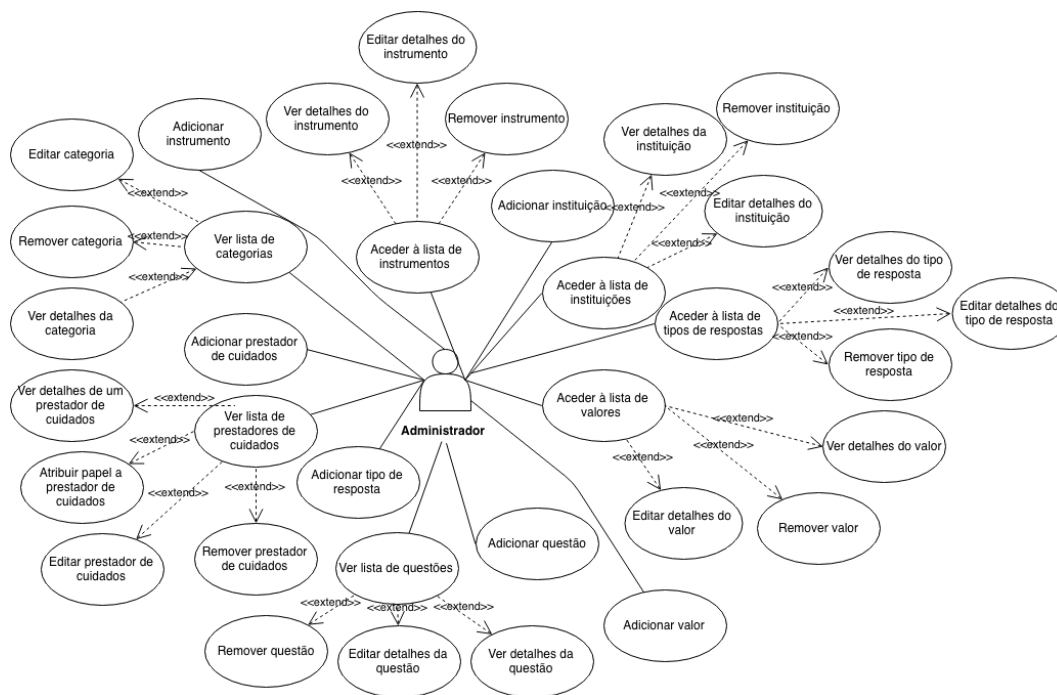


Figura 3.2: Diagrama de casos de uso do Administrador

A tabela 3.1 apresenta o nome do caso de uso, os atores que nele intervêm e uma descrição breve.

Caso de uso	Atores	Descrição
Registo	Prestador de Cuidados	O prestador de cuidados poderá registar-se no sistema preenchendo um formulário com os seus dados pessoais, tais como nome, e-mail, data de nascimento, telefone, género, morada e palavra passe.
Iniciar/ Terminar Sessão	Prestador de Cuidados, Administrador	Para iniciar sessão, o utilizador terá que inserir o seu e-mail e palavra passe. No caso de a entrada relativa ao e-mail e o código hash relativo à palavra passe coincidirem com um utilizador previamente registado, o utilizador terá então acesso à plataforma. Com a sessão iniciada, e só nesse caso, existe então a opção de a terminar sessão.

Adicionar Avaliação	Prestador de Cuidados	O prestador de cuidados poderá adicionar uma nova avaliação que já tenha sido efetuada ou que esteja a ser efetuada no momento. Para tal, existirá a opção “Adicionar Avaliação” no menu “Avaliações”, que visa permitir selecionar a data da avaliação, no caso de a data em que o teste tenha sido realizado não seja a mesma da catalogação das respostas do mesmo, o utente a quem a avaliação pertence, através do nome completo, a instituição na qual a avaliação foi feita e o instrumento utilizado na avaliação. Depois de estes dados serem selecionados, será possível avançar para o preenchimento das questões presentes no instrumento selecionado.
Preencher avaliação	Prestador de Cuidados	Depois de o prestador de cuidados submeter os dados relativos à avaliação na página “Adicionar Avaliação”, este poderá submeter as respostas a cada uma das perguntas do questionário. Nesta página serão apresentadas as perguntas, ordenadas pelo número da pergunta, e as possíveis respostas. O utilizador a catalogar a avaliação poderá selecionar a pontuação dada a cada questão e submeter, apenas quando todas estiverem preenchidas.
Aceder à lista de Avaliações	Prestador de Cuidados	O prestador de cuidados terá acesso à lista de avaliações efetuadas tanto por si mesmo, como por outros utilizadores, através da opção “Ver Avaliações” no menu “Avaliações”. Esta lista deverá conter os nomes dos utentes, o nome do profissional responsável, a data, o nome do instrumento utilizado e, dependendo do instrumento, a pontuação total do questionário em cada avaliação. Caso o utilizador tenha realizado a avaliação, para além dos campos anteriormente mencionados, serão apresentados três ícones que permitem editar, ver detalhes e remover a avaliação, caso contrário, só será apresentado o ícone de ver detalhes.

Ver Detalhes da Avaliação	Prestador de Cuidados	Depois de selecionar uma avaliação, o prestador de cuidados terá acesso às informações relativas a essa avaliação. Os detalhes incluem a data de realização e de catalogação da avaliação; o nome da instituição onde a avaliação foi feita; os dados do utente, tal como nome completo, data de nascimento, telefone e email (caso estes constem no perfil do utente); informação sobre o instrumento utilizado, tal como o nome do mesmo e a categoria em que se insere e o nome e email do utilizador que executou a avaliação. O prestador de cuidados terá acesso à avaliação executada. Serão apresentadas as perguntas do instrumento e suas respostas numa tabela que permite a edição dos valores submetidos como resposta em “Editar Respostas” ou a remoção dos mesmos, em “Remover respostas”. Caso o instrumento tenha uma pontuação total com base nas respostas dadas, esta será apresentada no final.
Remover respostas de Avaliação	Prestador de Cuidados	A pergunta e resposta submetida serão apresentadas de modo a confirmar a remoção das mesmas da avaliação. O prestador de cuidados poderá carregar no item “Excluir” para concluir a operação ou ir para a página anterior, “Ver Avaliações”.
Editar respostas de Avaliação	Prestador de Cuidados	O prestador de cuidados terá acesso a uma página que apresenta a pergunta selecionada por extenso e permite selecionar um valor de todos os valores possíveis de resposta, permitindo assim modificar a resposta submetida anteriormente. Existe um ícone que permitirá submeter a nova resposta à pergunta e um que permitirá voltar à página anterior.
Editar detalhes de Avaliação	Prestador de Cuidados	A página de edição dos detalhes da avaliação permitirá trocar algum dos dados da avaliação. Será possível alterar a data em que a avaliação foi executada, a data de catalogação, selecionar outro utente de uma lista que contém os nomes de todos os utentes e mudar o prestador de cuidados que realizou a avaliação. Contudo, não é possível mudar o instrumento escolhido para a avaliação. O prestador de cuidados pode submeter os novos dados para que estes sejam guardados ou voltar à página anterior, sem modificar nenhum dos detalhes.

Remover Avaliação	Prestador de Cuidados	Quando o prestador de cuidados selecionar a avaliação a remover, os dados da mesma serão exibidos na página com a opção de remover definitivamente ou de voltar atrás, para a os detalhes da avaliação.
Adicionar Utente	Prestador de Cuidados	Ao selecionar “Adicionar Utente” no menu de utentes, o prestador de cuidados poderá adicionar os dados de um utente: nome completo, e-mail, data de nascimento, título, telefone, género e escolaridade. Após preencher os campos necessários, será possível submeter para então criar um novo utente na base de dados ou ir para a página anterior.
Ver lista de Utentes	Prestador de Cuidados	O prestador de cuidados terá acesso à lista de utentes presentes no sistema através da opção “Ver Utentes” no menu de “Utentes”. Será apresentada uma tabela que contém certos dados dos utentes como nome, data de nascimento e contacto. Para cada utente, será possível selecionar um dos botões “Ver Detalhes”, “Editar Utente” e “Remover Utente”. No caso de o prestador de cuidados procurar por um utente em específico, existirá uma barra de procura onde poderá ser inserida parte do nome do utente e todos os utentes cujos nomes contém as palavras chave inseridas serão enumerados com as opções mencionadas anteriormente.
Editar detalhes de um utente	Prestador de Cuidados	Depois de selecionar a edição de um utente em concreto, o prestador de cuidados poderá modificar quaisquer dados que tenham sido inseridos ou até mesmo adicionar campos que estivessem vazios. Será possível submeter carregando no botão para tal efeito ou voltar à página anterior.
Adicionar utilizadores	Administrador	O administrador poderá aceder à página “Adicionar Utilizador” através do menu “Utilizadores”. Existirão vários campos para preencher, nomeadamente nome completo, data de nascimento, género, email, password, número de telefone e papel no sistema.

Atribuir papel a utilizador	Administrador	Depois de um utilizador se registar, o administrador pode atribuir-lhe um papel caso queira garantir acesso ao sistema. Para isto, basta seleccionar o utilizador a quem quer atribuir o papel, clicar no botão "Editar" e seleccionar um dos papéis presentes no campo "Papel do Utilizador". Após preencher o campo, pode-se submeter através do botão para continuar ou voltar para trás e cancelar a operação.
Ver lista de utilizadores	Administrador	É possível ver a lista de utilizadores registados no sistema através do menu "Utilizadores", opção "Ver lista de Utilizadores". A informação será apresentada num formato de tabela com os campos: nome completo, data nascimento, papel e e-mail. Em cada linha, existem os botões com as opções de editar, ver detalhes ou eliminar o utilizador.
Editar utilizador	Administrador	Depois de um utilizador se registar, o administrador deverá editar os seus detalhes. Para isto, terá de seleccionar o utilizador a quem quer atribuir o papel, clicar no botão "Editar" e alterar os campos pretendidos. Após alterar um ou mais campos, pode-se submeter as alterações através do botão para continuar ou voltar para trás e cancelar a operação.
Remover utilizador	Administrador	Dado um utilizador registado, o administrador poderá clicar no botão com o ícone de remover que se encontra ao lado do nome do utilizador que pretende eliminar.
Ver lista de instituições	Administrador	No menu "Tabelas", será possível seleccionar a opção "Instituições", que irá apresentar a lista de instituições em forma de tabela, apresentado assim os dados das instituições. Existirá um campo de busca que permitirá pesquisar por nome da instituição ou cidade e a informação está dividida em várias páginas. Para cada uma das instituições, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existe um botão com o ícone "Adicionar" para adicionar instituições.

Adicionar instituição	Administrador	Depois de carregar no botão "Adicionar" na página que lista as instituições, será apresentada uma página com os campos a preencher com os detalhes da instituição, tais como nome, morada, código postal, cidade e tipo de instituição. Depois de preencher os campos obrigatórios, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.
Editar instituição	Administrador	A página de edição de uma instituição permitirá que o valor dos campos pertencentes à instituição sejam editados ou que se preencham campos que não foram anteriormente preenchidos por não serem obrigatórios.
Remover instituição	Administrador	A página de remoção de instituição apresentará os dados relativos à instituição que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. Uma instituição só poderá ser eliminada se não existirem avaliações a ela associadas.
Ver detalhes da instituição	Administrador	Quando o administrador escolher "Ver Detalhes" de uma instituição, uma página com todos os campos relativos à instituição será apresentada. Existirão dois botões no final da página, um para voltar para trás e outro para editar os detalhes que estão a ser apresentados.
Ver lista de instrumentos	Administrador	No menu "Tabelas", será possível selecionar a opção "Instrumentos", que apresentará a lista de instrumentos presentes no sistema em forma de tabela, mostrando assim os dados de cada instrumento. Existirá um campo de busca que permitirá pesquisar por nome do instrumento ou pela categoria a que ele pertence e a informação encontra-se à dividida em várias páginas. Para cada um dos instrumentos, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existirá um botão com o ícone "Adicionar" que irá permitir adicionar novos instrumentos.
Adicionar instrumento	Administrador	Depois de carregar no botão "Adicionar" na página que lista os instrumentos, será apresentada uma página com os campos a preencher com os detalhes do instrumento, tais como nome, categoria, descrição e total, se for caso disso. Depois de preencher os campos obrigatórios, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.

Editar Instrumento	Administrador	A página de edição de um instrumento permitirá que o valor dos campos a este associados sejam editados ou que sejam preenchidos, no caso de não serem obrigatórios.
Remover instrumento	Administrador	A página de remoção de instrumentos apresentará os dados relativos ao instrumento que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. Um instrumento só poderá ser eliminado se não houverem avaliações a ele associadas no entanto, se esta condição se verificar e o instrumento tiver questões a ele associadas, estas serão eliminadas.
Ver detalhes do instrumento	Administrador	Quando o administrador escolher ver os detalhes de um instrumento, uma página com todos os campos relativos a este será apresentada. Existem dois botões no final da página, um para voltar para trás e outro para editar os detalhes que estão a ser apresentados.
Ver lista de categorias	Administrador	No menu "Tabelas", será possível selecionar a opção "Categorias", que apresentará a lista de categorias que existem no sistema em forma de tabela, mostrando assim os dados de cada categoria. Para cada uma das categorias, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existirá um botão com o ícone "Adicionar" que permite adicionar novas categorias.
Adicionar categoria	Administrador	Depois de carregar no botão "Adicionar" na página que lista as categorias, será apresentada uma página com os campos a preencher com os detalhes dessa mesma categoria, tais como nome e descrição, sendo esta última opcional. Depois de preencher os campos obrigatórios, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.
Editar categoria	Administrador	A página de edição de uma categoria permitirá que os valores inseridos nos campos relativos à categoria sejam editados ou preenchidos, no caso de ainda não se encontrarem preenchidos por não serem obrigatórios.
Remover categorias	Administrador	A página de remoção de categorias apresentará os dados relativos à categoria que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. Uma categoria só poderá ser eliminada se não existirem instrumentos a ela associados.

Ver detalhes da categoria	Administrador	Quando o administrador escolher ver os detalhes de uma categoria, todos os campos e atuais valores serão apresentados. Existirão dois botões no final da página, um para voltar para trás e outro para editar os detalhes que estão a ser apresentados.
Ver lista de tipo de resposta	Administrador	No menu "Tabelas", será possível selecionar a opção "Tipo de Resposta", que irá apresentar a lista dos tipos de resposta que existem no sistema em forma de tabela. Para cada uma dos elementos apresentados, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existirá um botão com o ícone "Adicionar" que permite adicionar novos tipos de respostas. Existirá ainda um campo de busca que visa permitir pesquisar por nome do tipo de resposta e a informação será dividida em várias páginas, que possibilitam a navegação.
Adicionar tipo de resposta	Administrador	Depois de carregar no botão "Adicionar" na página que lista os tipos de resposta presentes no sistema, será apresentada uma página com o campo que é necessário preencher com os detalhes desse mesmo tipo de resposta, neste caso o nome, Depois de preencher o campo obrigatório, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.
Editar tipo de Resposta	Administrador	A página de edição de um tipo de resposta apresentará o valor que a mesma tem atualmente, e permitirá que este seja editado.
Remover tipo de Resposta	Administrador	A página de remoção do tipo de resposta apresentará os dados relativos ao tipo de resposta que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. Só se poderá eliminar um tipo de resposta depois de eliminar os valores a ele associados, isto pode ser feito nos detalhes do tipo de resposta.

Ver detalhes do tipo de Resposta	Administrador	Quando o administrador escolher ver os detalhes de um tipo de resposta, uma página com o nome e os valores associados a este tipo de resposta será apresentada. Será possível adicionar um valor a este tipo de resposta através do ícone "Adicionar" ou remover carregando na cruz que aparece ao lado dos valores, sendo necessário confirmar esta ação posteriormente. Existirão dois botões no final da página, um para voltar para trás e outro para editar os detalhes que estão a ser apresentados.
Ver lista de valores	Administrador	No menu "Tabelas", será possível selecionar a opção "Valores", que apresentará a lista de valores que existem no sistema em forma de tabela, mostrando assim os dados de cada elemento. Para cada um dos valores, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existirá um botão com o ícone "Adicionar" que permite adicionar novos valores ao sistema. Existirá também um campo de busca que permite pesquisar por valor e a informação encontra-se dividida em várias páginas, sendo possível navegar entre elas através do contador de páginas no topo.
Adicionar valor	Administrador	Depois de carregar no botão "Adicionar" na página que lista os valores presentes no sistema, será apresentada uma página com o campo que será necessário preencher com o valor. Depois de preencher o campo obrigatório, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.
Editar valores	Administrador	A página de edição de um valor apresentará o valor atual, e permitirá que este seja editado.
Remover valor	Administrador	A página de remoção do valor apresentará o valor que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. Este valor não pode estar associado a nenhuma avaliação.
Ver detalhes do valor	Administrador	Quando o administrador escolher ver os detalhes de um valor, uma página com esse valor será apresentada. Existirão dois botões no final da página, um para voltar para trás e outro para editar este campo.

Ver lista de Roles	Administrador	No menu "Utilizadores", será possível selecionar a opção "Papéis", que apresentará a lista de papéis que existem no sistema em forma de tabela. Para cada um dos papéis, existirão botões com ícones "Editar", "Ver detalhes" e "Remover". No início da página existirá um botão com o ícone "Adicionar" que permite adicionar novos papéis ao sistema. Existirá também um campo de busca que permite pesquisar por nome do papel de utilizador e a informação encontra-se dividida em várias páginas, sendo possível navegar entre elas através do contador de páginas no topo.
Adicionar role	Administrador	Depois de carregar no botão "Adicionar" na página que lista os papéis de utilizador presentes no sistema, será apresentada uma página com o campo que é necessário preencher com o nome relativo ao papel. Depois de preencher o campo obrigatório, pode-se carregar no botão de submeter ou voltar para trás e cancelar a operação.
Editar Role	Administrador	A página de edição de um valor permite que o mesmo seja editado.
Remover Role	Administrador	A página de remoção do papel apresentará o elemento que se pretende eliminar e dois botões, um para prosseguir e concluir a operação e outro para cancelar e voltar à página anterior. O papel só pode ser removido se não estiver atribuído a nenhum utilizador.
Atribuir um prestador de cuidados a um utente	Prestador de Cuidados	Um prestador de cuidados com o papel "Gestor de caso" poderá fazer a relação entre o utente e os seus prestador de cuidados.

Tabela 3.1: Tabela de Descrição dos casos de uso

3.6 Mockups

No sentido de melhor visualizar o sistema a ser desenvolvido para cumprir os objetivos da presente dissertação, foram criados mockups com uma possível interface contemplando a organização de informação em diferentes páginas, sendo estas: criação, edição, remoção e ver detalhes. Para os instrumentos e seus componentes, avaliações, utentes e utilizadores, terão que existir páginas que permitam a criação, edição e remoção dos mesmos. As Figuras 3.3, 3.4 e 3.5 apresentam mockups da estrutura de páginas de criação, edição e remoção de instrumentos, neste caso em específico.

Início

Sobre

Contato

Tabelas

Utilizador

[Conta](#) [Terminar sessão](#)

Adicionar Instrumento

Nome

Categoria

Categoria

Total

Descrição (opcional)

Cancelar

Adicionar

Figura 3.3: Mockup de uma página de criação de um novo instrumento

As páginas de criação deverão incluir todos os campos a ser preenchidos pelo utilizador ou administrador, que, neste caso, serão os campos relativos ao instrumento: o nome, a categoria, selecionada a partir da lista de categorias já inseridas no sistema, o total do instrumento, caso exista, e uma descrição opcional.

Início

Sobre

Contato

Tabelas

Utilizador

[Conta](#) [Terminar sessão](#)

Editar Instrumento

Nome

Índice de Pfeffer

Categoria

Física

Total

Descrição (opcional)

Constituído por itens que classificam a capacidade do indivíduo para realizar AIVD e funções cognitivas/sociais como fazer compras, preparar refeições, manter-se atualizado, prestar atenção a programas de rádio ou TV e discuti-los.

Cancelar

Alterar

Figura 3.4: Mockup de uma página de edição de instrumentos

As páginas de edição deverão incluir todos os campos do elemento e o valor previamente inserido pelo ator do sistema numa caixa de texto editável, permitindo assim a edição ou alteração destes dados. Os campos presentes nesta página serão os mesmos da página de criação do instrumento.

Início Sobre Contacto Tabelas Utilizador [Conta](#) [Terminar sessão](#)

Remover Instrumento

Nome Índice de Pfeffer

Categoria Física

As questões associadas a este instrumento também serão eliminadas.
Pretende proceder?

Figura 3.5: Mockup de uma página de remoção de um instrumento

No caso das páginas de remoção de elementos do sistema, estas deverão incluir todos os campos obrigatórios do elemento em questão. Por exemplo, no caso do instrumento, será apresentado o nome e a categoria, uma vez que o total e descrição não são campos obrigatórios. Isto permite que o administrador confirme que é este o instrumento que quer eliminar. Em adição, no caso de haverem avaliações associadas a este instrumento, aparecerá uma mensagem de erro, como se pode ver na Figura 3.6, no entanto, será possível eliminar instrumentos mesmo que estes tenham questões associadas. Esta informação será apresentada ao administrador para que este confirme se pretende eliminar o instrumento, mesmo que isso implique eliminar todas as questões a ele associadas.

Não é possível eliminar este instrumento
porque existem avaliações a ele associadas.

Figura 3.6: Mensagem de erro que aparece quando o administrador tenta remover um instrumento com avaliações associadas

Para além das páginas já mencionadas, serão necessárias páginas que permitam ver todos os elementos em formato de lista e páginas que permitam ver os detalhes individuais de cada um. Neste sentido, a Figura 3.7 apresenta um mockup de uma página que irá elencar todos os instrumentos, pela ordem em que estes foram adicionados no sistema, e que terá uma barra de pesquisa para facilitar a procura de um determinado instrumento, quer através de parte do seu nome, quer através da categoria a que ele pertence.

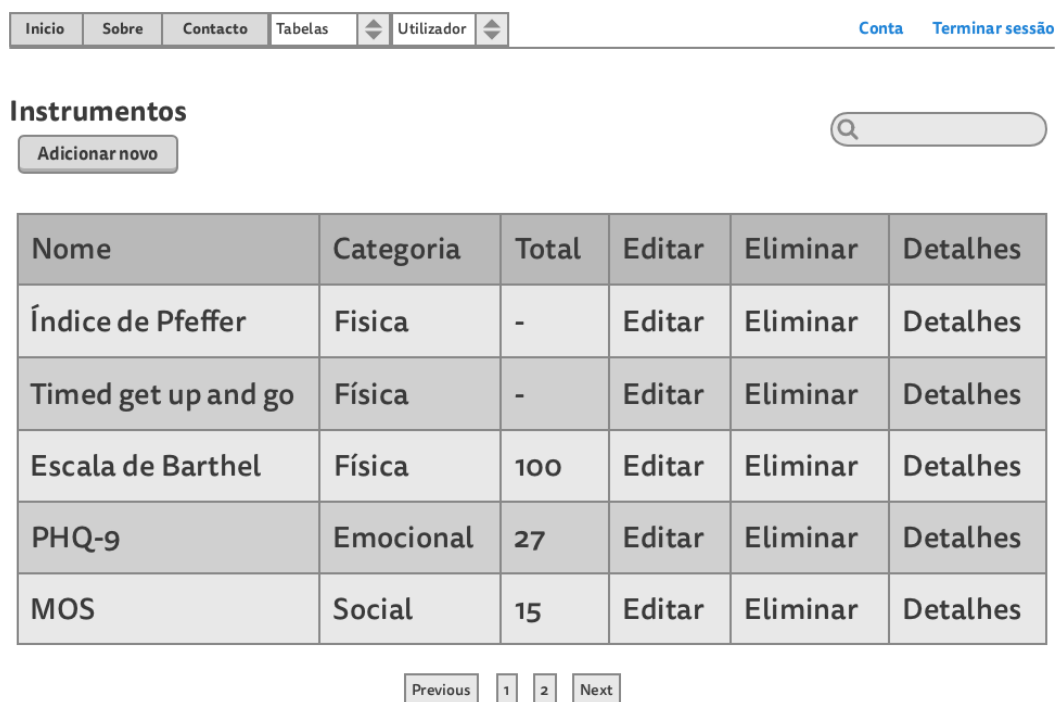


Figura 3.7: Mockup de uma página que enumera todos os instrumentos e permite procurar um instrumento específico

Como a lista de instrumentos disponíveis para este sistema é extensa, existirá também uma barra de navegação que permitirá navegar na lista de dez em dez elementos. Esta barra não aparecerá caso existam menos de dez instrumentos no sistema. Esta barra de navegação estará presente para todos os elementos cuja página de enumeração contenha uma lista extensa de elementos, havendo a exceção da página principal relativa às categorias do sistema, que só contém cinco elementos.

Adicionalmente, será necessária uma página para permitir ver os detalhes de um determinado elemento do sistema, a qual terá de incluir todos os campos associados, bem como outros elementos que o constituam. A título exemplificativo, apresenta-se na Figura 3.8, uma mockup específica para uma página de detalhes de um instrumento, onde terão que ser apresentados todos os campos relativos ao mesmo: nome, categoria, total, se este existir, e descrição. Em adição, como um instrumento poderá conter várias questões, estas também

serão apresentadas.

Início

Sobre

Contato

Tabelas

Utilizador

Conta Terminar sessão

Detalhes do instrumento

Nome	Índice de Pfeffer
Categoria	Física
Descrição (opcional)	Constituído por itens que classificam a capacidade do indivíduo para realizar AIVD e funções cognitivas/sociais como fazer compras, preparar refeições, manter-se atualizado, prestar atenção a programas de rádio ou TV e discuti-los.

Cancelar

Editar

Nº	Questão
1	A pessoa idosa é capaz de cuidar do seu próprio dinheiro?
2	A pessoa idosa é capaz de fazer as compras sozinho (por exemplo de comida e roupa)?
3	A pessoa idosa é capaz de esquentar água para café ou chá e apagar o fogo?
4	A pessoa idosa é capaz de preparar comida?
5	A pessoa idosa é capaz de manter-se a par dos acontecimentos e do que se passa na vizinhança?
6	A pessoa idosa é capaz de prestar atenção
7	A pessoa idosa é capaz de lembrar de compromissos e acontecimentos familiares?
8	A pessoa idosa é capaz de cuidar de seus próprios medicamentos?
9	A pessoa idosa é capaz de andar pela vizinhança e encontrar o caminho de volta para casa?
10	A pessoa idosa é capaz de cumprimentar seus amigos adequadamente?
11	A pessoa idosa é capaz de car sozinho (a) em casa sem problemas?

Figura 3.8: Mockup de uma página que permite ver os detalhes de um instrumento

Os mockups apresentados são relativos aos instrumentos, no entanto, terão que existir páginas com formatos semelhantes para as mesmas funções relativas aos elementos que constituem os instrumentos, avaliações, utilizadores ou até mesmo utentes. A página de detalhes do utente será, contudo, uma exceção, uma vez que será um dashboard interativo como proposto nos objetivos da presente dissertação. Na Figura 3.9, é apresentado o

mockup com a estrutura de uma dashboard que permita visualizar: os dados do paciente, as avaliações a este executadas e gráficos dinâmicos que permitam relacionar os resultados das avaliações correspondentes ao mesmo instrumento de modo a perceber se houve alterações significativas.

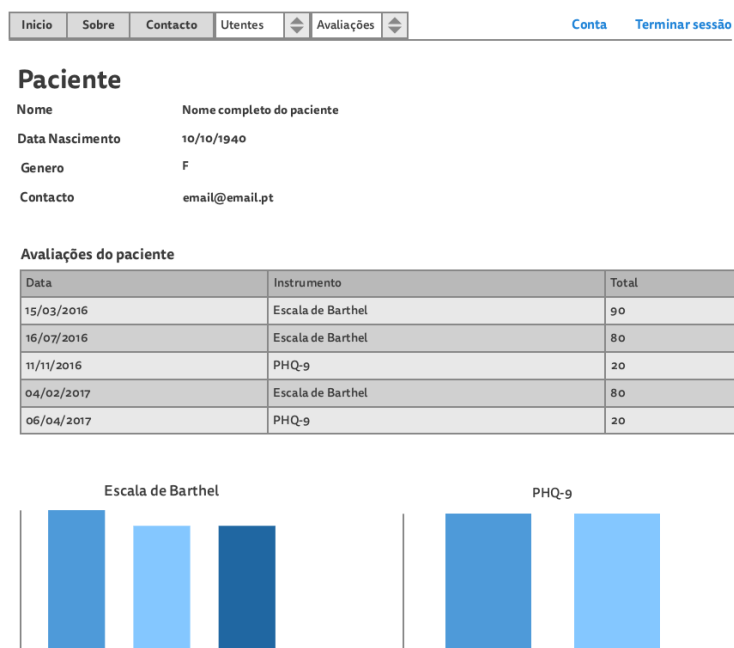


Figura 3.9: Mockup da página de detalhes de um utente

3.7 Conclusão

Os requisitos funcionais e não funcionais, em conjunto com os casos de uso apresentados no presente capítulo, servirão de base à implementação do sistema, a qual será discutida em profundidade no próximo capítulo.

Capítulo 4

Implementação

No capítulo 3 foi explicado o conceito, os atores, os requisitos e os use cases da aplicação. Neste capítulo, serão abordados os temas e passos relevantes da implementação da arquitetura geral apresentada na secção anterior, incluindo as tecnologias utilizadas, aspetos específicos da arquitetura do sistema, o diagrama da Base de Dados (BD) e a descrição da implementação do sistema.

4.1 Tecnologias Utilizadas

Tendo em conta o estudo de diferentes tecnologias e arquiteturas descrito no capítulo 2, é possível tomar uma decisão quanto às mais adequadas para o desenvolvimento de uma plataforma de avaliação multidimensional da funcionalidade de pessoas idosas, tendo em conta os requisitos da mesma.

Para cumprir os objetivos desta dissertação, foi utilizada uma arquitetura Model-View-Controller (MVC), que se baseia na divisão entre Controlador, Modelo e View. Ao contrário das outras arquiteturas estudadas, esta arquitetura permite que a informação passe entre os três componentes sem necessidade de interfaces e é eficiente mesmo em caso de não existir contexto entre a View e os outros componentes. Dentro das frameworks estudadas que suportam esta arquitetura, a que permite fazer uma melhor gestão e ligação entre a parte da visualização da plataforma e o servidor é a ASP.NET MVC 6.

Uma aplicação ASP.NET MVC 6 utiliza a linguagem C Sharp nos Controladores para tratar da interação com o utilizador, fazer queries à BD utilizando a Structured Query Language (SQL), comunicar com o Modelo e selecionar a View a ser apresentada ao utilizador. As views são ficheiros com a extensão .html, consistindo assim em ficheiros HyperText Markup Language (HTML) que permitem o uso de código C Sharp que faz a ligação entre estas e os controladores. Em adição, o ASP.NET MVC utiliza um gestor de pacotes NuGet que disponibiliza diversas ferramentas e serviços necessários ao desenvolvimento de uma aplicação. Estes pacotes incluem o Bootstrap [51] e o jQuery [52]. O Bootstrap é uma framework que é responsável pelo desenvolvimento de componentes de interface e front-end de aplicações web, utilizando HTML, Cascading Style Sheets (CSS) e javascript para ga-

rantir que as páginas são interativas, esteticamente agradáveis, fáceis de usar e responsivas. Aquando o desenvolvimento desta dissertação, a versão 4 do Bootstrap foi disponibilizada. Como esta versão é uma grande mudança a nível de toda a apresentação do projeto e não é de migração direta, foi utilizado a versão 3.3.7. O jQuery 3.3.1. é uma biblioteca de javascript que pode ser utilizada para gestão de transações em documentos HTML, manipulação de eventos e animações e interações AJAX.

O desenvolvimento da aplicação também exigiu pacotes extra como o FontAwesome, que inclui ícones utilizados nas views, e o Microsoft ASP.NET Identity, que é utilizado para a gestão dos utilizadores e atribuição de papéis dentro da aplicação.

A nível de armazenamento dos dados, tendo em conta que seria necessário guardar os dados dos utilizadores do sistema, dos utentes e toda a informação relativa aos instrumentos utilizados e seu preenchimento, é possível concluir que se pode usar uma BD relacional. Para este efeito, foi escolhido o servidor de BD SQL Server por ser uma tecnologia compatível com a framework web selecionada.

4.2 Arquitetura do Sistema

No capítulo 2, foram estudadas diferentes modelos arquiteturais. Depois de uma análise, o mais adequado aos objetivos desta dissertação é a arquitetura MVC. Assim sendo, a arquitetura da aplicação, ilustrada na Figura 4.1, baseia-se numa estrutura de Modelo, Controlador e View ligado a uma BD SQL.

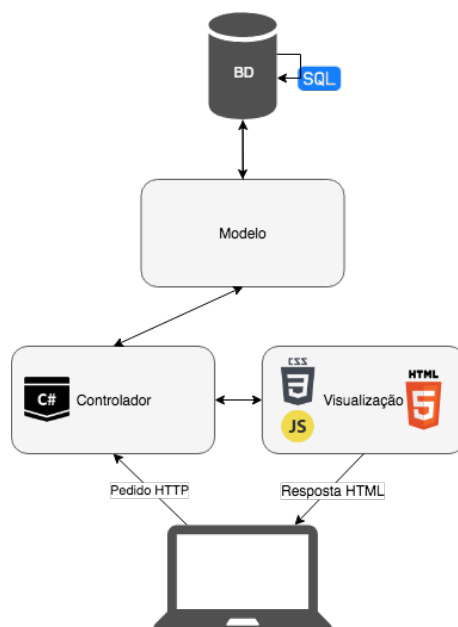


Figura 4.1: Arquitetura do Sistema

4.3 Base de Dados

Considerando os dados que têm que ser guardados, o tipo de BD selecionado foi o relacional, uma vez que se tratam de questionários com um formato fixo de pergunta-resposta. As questões desse questionário fazem parte de um instrumento, sendo que este está inserido numa categoria.

Para a informação que tem que ser armazenados na BD, existem diversas tabelas a considerar tanto a nível dos instrumentos a ser utilizados na avaliação como a nível dos dados do utilizador comum e utente, e por fim, dados relativos às avaliações executadas.

A figura 4.2 representa o diagrama de BD relativa à representação de modelo de dados utilizado nesta dissertação. Através da sua análise, é possível concluir que existem diversas tabelas, mas as que representam a estrutura principal são: "Avaliação", "Instrumento", "Utilizador", "Paciente" e "Tripleto_Resposta". As avaliações são definidas por um identi-

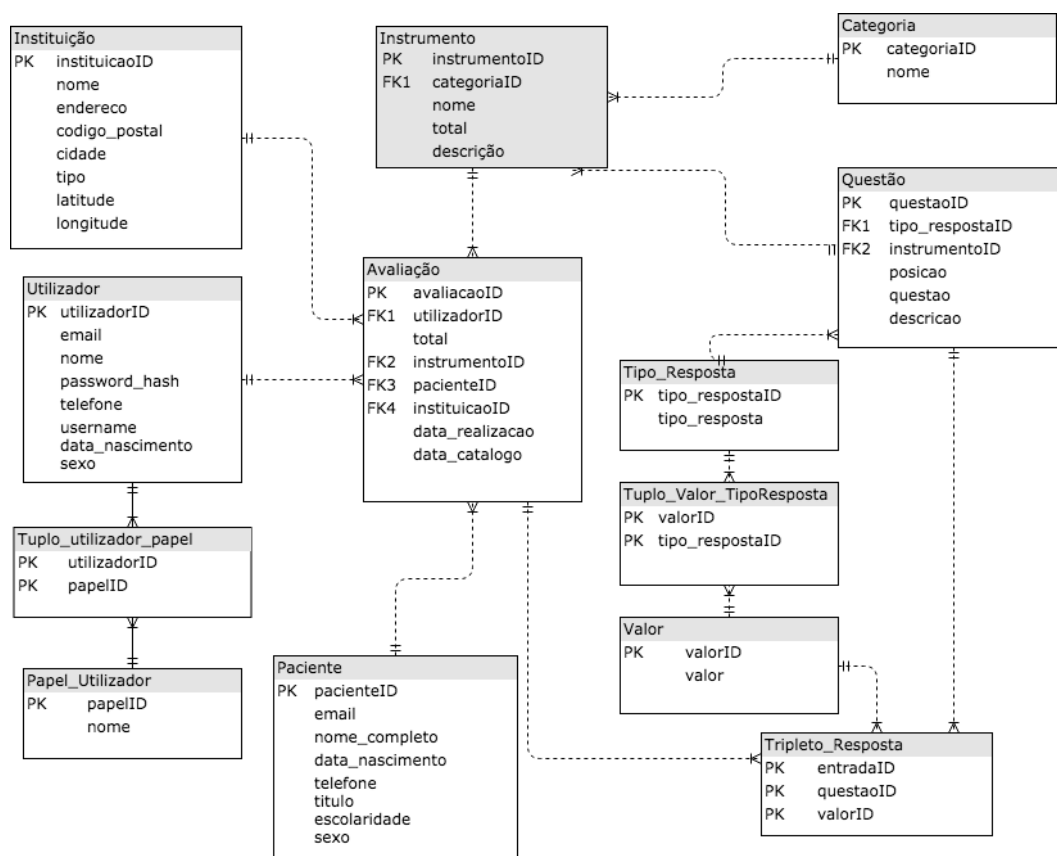


Figura 4.2: Diagrama da BD

ficador que será a sua chave primária e contém várias chaves estrangeiras que representam o instrumento a ser utilizado, o utilizador que executa a avaliação, o utente a quem a avaliação é feita e a instituição onde foi executada, para além dos campos das datas relativas à avaliação.

O instrumento, por sua vez, terá o seu próprio identificador como chave primária, e uma chave estrangeira relativa à categoria. Cada instrumento pode ter diversas questões. Cada questão terá o seu identificador e uma chave estrangeira referente ao instrumento e outra relativa ao tipo de resposta que é aceite naquela questão.

As tabelas "Pacientes", "Instituições", "Utilizadores", "Tipo_Resposta" e "Valor" são constituídas pelo seu identificador, que é uma chave primária, e os restantes campos relativos à informação associada.

Existem ainda duas tabelas de tuplos, sendo estes definidos como estruturas bidimensionais com determinado esquema e zero ou mais instâncias que definem a linha de uma relação: o "Tuplo_Valor_TipoResposta" e o "Tuplo_utilizador_papel". O tuplo que conecta o identificador do Tipo de Resposta com o Valor é utilizado porque diferentes tipos de respostas podem ter diversos valores. Do mesmo modo, o segundo tuplo que conecta o identificador do utilizador com o do papel que lhe é atribuído. Um utilizador pode ter vários papéis.

Por fim, a tabela "Tripleto_Resposta" representa as respostas às questões do instrumento escolhido. É identificada através da entrada, questão e valor associado.

4.4 Aplicação Web

Tal como foi referido anteriormente, a arquitetura MVC destaca-se por um formato que se baseia nos seguintes componentes: Modelo, Controlador e View. Estes três componentes comunicam entre si, e com a BD, de modo a criar uma aplicação coesa. Neste sentido, as três componentes estão interligadas entre si, como se pode ver através da arquitetura na Figura 4.1, na medida em que o Controlador apresenta e controla o estado da View, para além de gerir a navegação da plataforma. Em adição, o Controlador também pode atualizar o Modelo, ou pedir a este que se atualize a ele mesmo, e essas atualizações serão apresentadas na parte da View.

4.4.1 Object-Relational Mapping

Object-Relational Mapping (ORM) é uma técnica de desenvolvimento que permite mapear as estruturas de uma BD relacional (por exemplo, SQL Server, Oracle ou MySQL, entre outros) utilizando uma linguagem de programação orientada ao objeto [53]. As tabelas presentes na BD são representadas através de classes e as colunas de cada tabela são representados como instâncias das classes correspondentes. Isto permite automatizar a conversão entre objeto para tabela e vice-versa.

O ORM apresenta uma vantagem comparativamente com outras técnicas tradicionais de conexão entre a linguagem orientada ao objeto e BD relacional, que é a redução da quantidade de código a ser escrita [54]. Isto acontece porque em vez de aceder à BD com os comandos em linguagem SQL, usa-se uma interface de programação simples que executa todo o trabalho de persistência.

Por outro lado, usar as ferramentas ORM implica um nível muito alto de abstração que

pode mascarar o que está a acontecer no código de implementação. Assim, depender demasiado do ORM pode resultar em BDs com um design incorreto [55].

O ORM utilizado para .NET é o Entity Framework 6 (EF6), que permite o acesso às estruturas da BD como se estes fossem objetos .NET. O EF6 permite dois modos para começar um projeto ASP.NET:

- **Code First**

A abordagem Code First baseia-se em ter controlo no código, não na actividade da BD. Neste caso, as operações executadas sobre a BD provêm do código e as mudanças feitas manualmente à BD serão perdidas e é necessário usar entidades específicas, Plain Old CLR Objects (POCO), como modelo de dados [56].

- **Database First**

A abordagem Database First é utilizada quando já existe uma BD. Neste caso, a EF6 cria as entidades POCO automaticamente e, no caso de a BD ser atualizada, é possível atualizar o Modelo com base na BD [56].

A abordagem utilizada no âmbito desta dissertação foi a Database First, uma vez que o primeiro passo foi a criação da BD e só depois é que se passou para o desenvolvimento de uma aplicação web. Assim, usando a ferramenta de Scaffolding presente no Microsoft Visual Studio, é possível gerar código que permite a criação automática de código para exibir, criar, editar e eliminar dados [57].

Utilizando Asp.Net Identity User Profiles podem ser integradas contas com autenticação individual de utilizadores numa BD pré existente, como é o caso.

Cada elemento das tabelas presentes na BD terá um Controlador com uma ou mais views e um Modelo associado que permite o controlo desses dados em específico. Assim, numa primeira fase, a aplicação web funciona como uma interface da BD existente.

4.4.2 Modelo

Como já foi referido anteriormente, o Modelo, numa aplicação MVC, é referente à implementação da lógica adequada para o domínio de dados. Em termos práticos, o Modelo define as regras de apresentação de conteúdo nas Views da aplicação. Para permitir a existência de contas individuais com autenticação para utilizadores, por exemplo, existe um Application User que é definido num modelo específico de identidade, como se pode ver através da imagem 4.3. Os campos definidos, "NomeCompleto", "DataNascimento", "Género", "Morada" e "PhoneNumber" (Número de Telefone) são adicionados ao email e palavra passe que já estão incluídos na identidade do utilizador.

Os Modelos definem o formato da informação apresentada e a ser aceite nas views e quais os campos presentes em cada uma das tabelas da BD nos Controladores. Deste modo, todos terão um formato semelhante ao da Figura 4.4. No caso específico apresentado, trata-se do modelo da avaliação, contendo assim todos os campos que a constituem, bem como o seu formato e tipo e ainda a informação para ser utilizada nas Views. Assim, definem-se os diferentes campos, um por um. O campo de identificação da avaliação, que é único para

```

[Display(Name = "Nome Completo")]
[Required(ErrorMessage = "O nome completo é um campo obrigatório.")]
8 references
public string NomeCompleto { get; set; }
[Display(Name = "Data Nascimento")]
[DataType(DataType.Date)]
[DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:dd/mm/yyyy}")]
[Required(ErrorMessage = "A data de nascimento é um campo obrigatório.")]
7 references
public DateTime DataNascimento { get; set; }
[Display(Name = "Genero")]
[Required(ErrorMessage = "O genero é um campo obrigatório.")]
7 references
public string Sexo { get; set; }
[Display(Name = "Telefone/Telemóvel")]
[RegularExpression("[0-9]+", ErrorMessage = "Por favor introduza um número válido.")]
8 references
public string PhoneNumber { get; set; }

```

Figura 4.3: Modelo de Utilizador da Aplicação

cada uma das avaliações, "entryID" é um inteiro. No caso do identificador do utilizador que realiza e cataloga a avaliação, "userID", tem sempre como nome do campo "Prestador de Serviços", que é o que irá aparecer nas views que usam este modelo, e, em adição, o campo não permite que o campo de entrada relativo a este campo seja submetido vazio e caso aconteça, é imediatamente exibida uma mensagem de erro "Este campo é obrigatório". Para o caso da data, tanto de realização como de catalogação, uma vez que estas podem divergir, o tipo de dados tem que ser declarado como formato de data. A data real é um dado que tem que ser obrigatoriamente preenchido, contendo a mesma mensagem de erro que o identificador do utilizador, uma vez que a avaliação pode ter sido efetuada noutro dia e só estar a ser catalogada no momento. Por outro lado, a data de catalogação não é obrigatória porque não é introduzida pelo utilizador, o sistema guarda o valor da data em que a avaliação é submetida para ser guardada na BD. Tanto o identificador do instrumento como o do paciente são números inteiros obrigatórios, uma vez que só faz sentido fazer uma avaliação escolhendo o utente a avaliar e o instrumento adequado a utilizar. Ambas as páginas apresentam a mensagem de erro "Este campo é obrigatório." no caso de a entrada respetiva ser submetida vazia na interface. Por fim, no caso do total, trata-se de um campo opcional, dado que só alguns instrumentos é que possuem pontuação total, e é do tipo inteiro, assim como o identificador da instituição, "institutionID".

Todos os campos definidos no Modelo contém um "set" e um "get", dentro da sua declaração, o que vai permitir que os elementos sejam chamados no Controlador (get) para serem manipulados e alterados no Controlador e exibidos na View (set).

Todos os modelos presentes na aplicação seguem este padrão para as diferentes tabelas de dados utilizadas na BD, que são importadas e organizadas como Modelos. As únicas exceções são objetos modelo criadas para determinadas views como, por exemplo, as classes "theEntries" e "InstrumentEntriesPerPatient" que são exibidas na Figura 4.5. Neste caso, a classe "theEntries" referencia um instrumento e uma lista de avaliações. Este modelo de

```

public int entryID { get; set; }
[Display(Name = "Prestador de Serviços")]
[Required(AllowEmptyStrings = false, ErrorMessage = "Este campo é obrigatório.")]
4 references
public string userID { get; set; }
[DataType(DataType.Date)]
[Display(Name = "Data de Catalogação")]
[Required(ErrorMessage = "Este campo é obrigatório.")]
1 reference
public System.DateTime dataCatalog { get; set; }
[DataType(DataType.Date)]
[Display(Name = "Data de realização")]
4 references
public System.DateTime dataReal { get; set; }
[Display(Name = "Instrumento")]
[Required(AllowEmptyStrings = false, ErrorMessage = "Este campo é obrigatório.")]
4 references
public int instrumentID { get; set; }
[Display(Name = "Paciente")]
[Required(AllowEmptyStrings = false, ErrorMessage = "Este campo é obrigatório.")]
7 references
public int PatientID { get; set; }
[Display(Name = "Total")]
1 reference
public Nullable<int> total { get; set; }
[Display(Name = "Instituição")]
1 reference
public Nullable<int> institutionID { get; set; }

```

Figura 4.4: Modelo da avaliação

dados vai fazer parte do "InstrumentEntriesPerPatient", que possui uma lista com os instrumentos e as respectivas avaliações preenchidas e o identificador do utente. Isto permite que, no Controlador e na View, seja possível manipular e exibir estes dados, respetivamente.

Existem também vários modelos relativos à conta de utilizador, tanto de registo, alteração de palavra passe, login e as respetivas páginas (views) relativas à conta de utilizador, para que os campos sejam exibidos de acordo com o formato esperado e que as entradas cumpram obrigatoriamente os parâmetros exigidos para cada campo.

4.4.3 Controlador

O Controlador numa arquitetura MVC, especialmente na framework ASP.NET MVC, é responsável por fazer pedidos à BD (tanto de inserção como de recuperação de informação), retornar o Modelo a ser exibido na View e por toda a manipulação e alteração dos dados. Existe um Controlador para cada modelo de dados, bem como um para cada conjunto de páginas de criação, edição, remoção, visualização de detalhes e um índice.

O acesso à BD é feito através de queries. Em vez de escrever queries em SQL nativo, usa-se a linguagem Language Integrated Query (LINQ), que é um componente do .NET, para acesso à BD. Isto permite que os resultados da consulta da BD passem a objetos com contexto, caso não o fossem anteriormente.

Utilizando LINQ, a sintaxe dos métodos são compostas por métodos de extensão, incluídos ou na classe Enumerable ou Queryable, e expressões lambda.


```

public class theEntries
{
    1 reference
    public theEntries()
    {
        entries = new List<Entries>();
    }
    1 reference
    public Instrument instrument { get; set; }
    2 references
    public IList<Entries> entries { get; set; }
}
3 references
public class InstrumentEntriesPerPatient
{
    1 reference
    public InstrumentEntriesPerPatient()
    {
        entryInfo = new List<theEntries>();
    }
    2 references
    public IList<theEntries> entryInfo { get; set; }
    1 reference
    public Patient patient { get; set; }
}

```

Figura 4.5: Modelos desenvolvidos à medida

As queries utilizadas vão desde simples acessos a um modelo que é constituído apenas pela sua chave primária até operações complexas de utilização de dados que fazem parte do formulário da View.

```

Question thequestion = db.Question.Where(x => x.questionID == tripleto.questionID).FirstOrDefault();

```

Figura 4.6: Querie para extrair o objeto "Question" da BD

Na Figura 4.6, é apresentada a querie utilizada para, dado um identificador de uma questão ("tripleto.questionID"), obter o objeto "Question" da BD, com todos os seus campos. Assim, acede-se à BD (db na figura), tabela "Question", usando o método de extensão "Where" da classe Enumerable. A expressão lambda, neste caso, será a `x => x.questionID == tripleto.questionID`. Isto é um modo de seleccionar a questão cujo identificador corresponda ao pretendido. Por fim, o "FirstOrDefault" da linguagem LINQ serve para indicar que é para obter o primeiro elemento que satisfaça a condução do Where, caso este elemento exista.

Para editar ou remover os dados que já existem no sistema, é necessário seleccionar o

```

Entries entry = db.Entries.Find(entryid);
db.Entries.Remove(entry);
db.SaveChanges();
return RedirectToAction("Index");

```

Figura 4.7: Query para obter o objeto "Entry" pretendido da BD e eliminá-lo da mesma

objeto a editar ou eliminar. No exemplo da Figura 4.7, usa-se o método "Find" do DbSet, que usa o valor da chave primária para tentar encontrar uma entidade deste tipo, com este identificador, que já exista na BD. Para remover a entrada cujo identificador corresponde a um determinado parâmetro, usa-se `db.Entries` para aceder às avaliações armazenadas na BD e o `Remove(entry)`, onde a `entry` foi a avaliação encontrada através do Find executado anteriormente, remove a entrada permanentemente. Para que as alterações sejam efetuadas na BD, é necessário guardar estas alterações na BD com `db.SaveChanges()`. Este modelo tem ainda a característica de transformar todas as operações de um conjunto numa operação atómica, como se de uma só transação se tratasse.

```
int av = db.AnswerValues.Where(x => x.value.Equals(aux)).Select(a => a.valueID).FirstOrDefault();
```

Figura 4.8: Query para obter o inteiro guardado no campo valueID da tabela AnswerValues

Outro exemplo relativamente simples encontra-se na Figura 4.8 que, em vez de obter um objeto da BD como acontece no primeiro exemplo, vai mais além e seleciona um campo específico do objeto pretendido cujo identificador coincide com o pretendido. Para isto, depois do método "Where" onde se insere a expressão lambda relativa ao identificador do objeto que se pretende obter, usa-se o "Select" com a expressão lambda `a => a.valueID`, que seleciona o campo, neste caso "valueID", que será guardado na variável "av".

```
public ActionResult SaveChanges(int entryID, int questionID, string values)
{
    InstrumentAnswers theAnswer = db.InstrumentAnswers.Where(x => x.entryID == entryID && x.questionID == questionID).FirstOrDefault();
    AnswerValues answerValue = db.AnswerValues.Where(x => x.value == values).FirstOrDefault();
    ///--- Erase previous answer
    db.InstrumentAnswers.Remove(theAnswer);
    ///--- Create a new answer with the same entryID, questionID and the new valueID choosen by the user
    ///--- This was necessary because all the columns in this table are primary keys
    InstrumentAnswers newAnswer = new InstrumentAnswers();
    newAnswer.entryID = entryID;
    newAnswer.questionID = questionID;
    newAnswer.valueID = answerValue.valueID;
    db.InstrumentAnswers.Add(newAnswer);

    ///--- Save new answer
    db.SaveChanges();
    return RedirectToAction("Details", "Entries", new { id = entryID });
}
```

Figura 4.9: Função "SaveChanges" das avaliações

A função `SaveChanges` (Figura 4.9), que tem como propósito alterar uma resposta de uma das questões de uma avaliação que já tinha sido guardada previamente na BD. Para tal, encontra-se o triploto cujos identificadores da entrada e da questão correspondem aos argumentos da função, utilizando a expressão lambda `x => x.entryID == entryID && x.questionID == questionID` que define a condição de procura na tabela "InstrumentAnswers" da BD. Para aceder ao objeto correspondente ao valor da resposta, acede-se ao primeiro elemento, caso este exista, da tabela `AnswerValues`, onde o valor corresponde ao

que provém da interface sob a forma de argumento desta função.

Tal como já foi mencionado, para remover um campo de uma das tabelas da BD usa-se o `db`, que foi anteriormente declarado como a BD à qual a aplicação está ligada, seguida da tabela à qual o campo pertence, neste caso `InstrumentAnswers`, e remover o elemento, neste caso o objeto `InstrumentAndAnswers` que foi guardado anteriormente em `theAnswer`.

Para criar um novo objeto, é necessário instanciá-lo e declará-lo e só depois preenchê-lo adequadamente. No exemplo da Figura 4.9, cria-se um novo objeto através do `new InstrumentAnswers()` e só depois é que este será preenchido com os dados adequados. Para cada um dos campos do objeto `InstrumentAnswers`, é necessário declarar os novos valores, por exemplo `newAnswer.entryID = entryID`, e só depois adicionar este objeto à BD através de `db.InstrumentAnswers.Add(newAnswer)`.

Para que todas as alterações sejam guardadas na BD, é necessário evocar o método `SaveChanges()`.

4.4.4 View

As Views são responsáveis pela parte da interface e interação da aplicação com o utilizador. As views são páginas HTML com código C Sharp embutido para aceder aos dados enviados pelo Controlador. Em adição, também contêm javascript para manipulação das páginas.

Como já foi referido anteriormente, um administrador e um utilizador comum não têm o mesmo nível de acesso e a plataforma será substancialmente diferente, uma vez que os atores possuem funcionalidades muito diferentes.

Views Administrador

No caso do administrador, a barra de navegação terá o aspeto presente na Figura 4.10, uma vez que este só têm acesso às Tabelas, que constituem os diferentes objetos que integram todo o sistema de suporte à avaliação multidimensional da funcionalidade de pessoas idosas, incluindo todos os instrumentos, questões e possíveis respostas, e ao menu de utilizadores que permite gerir os utilizadores do sistema.



Figura 4.10: Barra de Navegação do Administrador

Para gerir o sistema em si, o menu "Tabelas", presente na Figura 4.11, inclui as seguintes opções: "Tipo de Resposta", que é referente aos tipos de resposta que são aceites para cada questão de cada instrumento; "Valores de Resposta", que são os valores possíveis para cada tipo de resposta, que irão aparecer durante o preenchimento da avaliação quando o utilizador selecionar qual deles é a resposta a submeter; "Categorias", que é a dimensão que é constituída por vários instrumentos e em princípio não será alterada, uma vez que só

existem as dimensões física, mental, psicológica, emocional e geral; Instrumentos; Questões e Instituições.

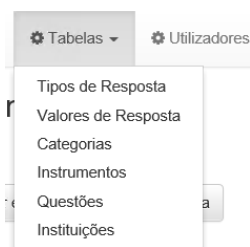


Figura 4.11: Menu de Gestão de Tabelas



Figura 4.12: Menu de Gestão de Utilizadores

Como se pode ver através da Figura 4.12, o menu "Utilizadores" inclui as opções de adicionar um utilizador, elencar os utilizadores ou elencar os papéis que os utilizadores podem ter.

Cada uma das opções do menu "Tabelas", bem como as do menu "Utilizadores" são constituídas por cinco views: "Índice", "Adicionar", "Remover", "Editar" e "Ver Detalhes". Apesar destas páginas associadas com Modelos e Controladores, o seu formato é sempre o mesmo, pelo que serão apresentados exemplos.

Tanto para gestão dos instrumentos e seus elementos como para os utilizadores, existe uma página de índice que apresenta todos os elementos. No caso dos instrumentos, por exemplo, apresenta todos os instrumentos e alguma informação. Por sua vez, no caso dos utilizadores apresenta todos os utilizadores e alguns dos seus detalhes. Isto é válido também para todos os outros elementos do menu de tabelas e de utilizadores.

Listas de elementos, como a apresentada na Figura 4.13, podem ser muito extensas e por isso os elementos são apresentados em páginas de dez elementos com uma barra de navegação na parte superior da página. Também é possível procurar por palavras chave relacionadas com um elemento através de uma barra de busca no canto superior direito. No caso da lista de instrumentos, por exemplo, a tabela de lista irá conter o nome de cada instrumento, a categoria a que pertence e o total, caso se justifique. Em cada linha, existem três ícones que remetem para três das restantes ações de acordo com o identificador do instrumento em causa: editar, ver detalhes e remover. Acima da tabela também existe um ícone para voltar à página anterior, que irá direcionar para a última página guardada no histórico, e um ícone de adição de novos elementos, que será outra das ações comuns a todos os elementos: Instrumentos, Questões, Tipos de Resposta, Valores, Categorias, Utilizadores e Papéis de Utilizador.

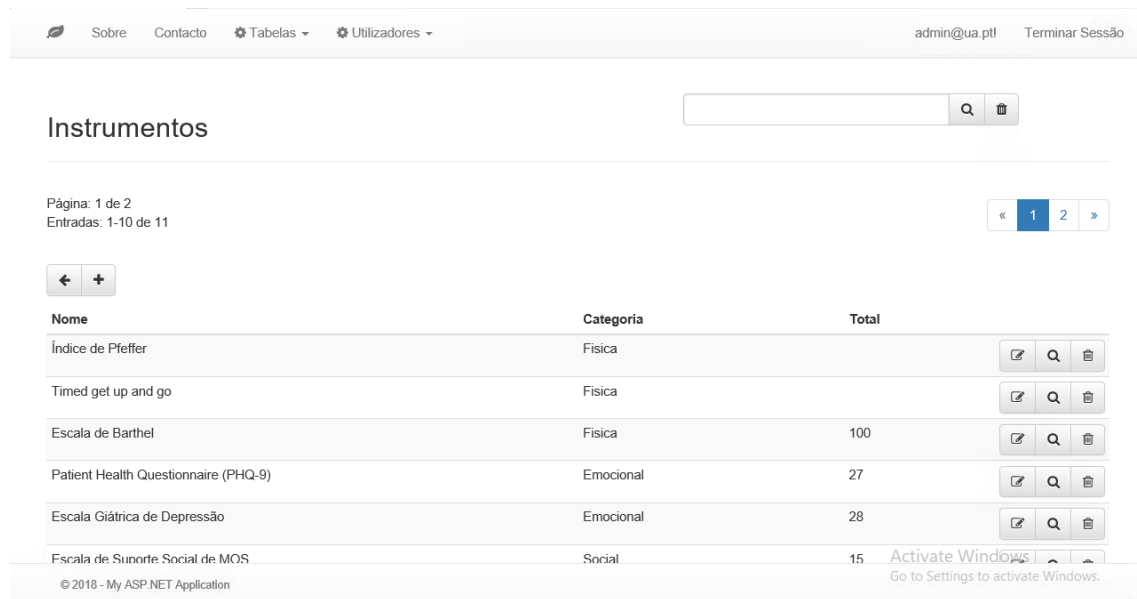


Figura 4.13: Página que contém a lista de instrumentos

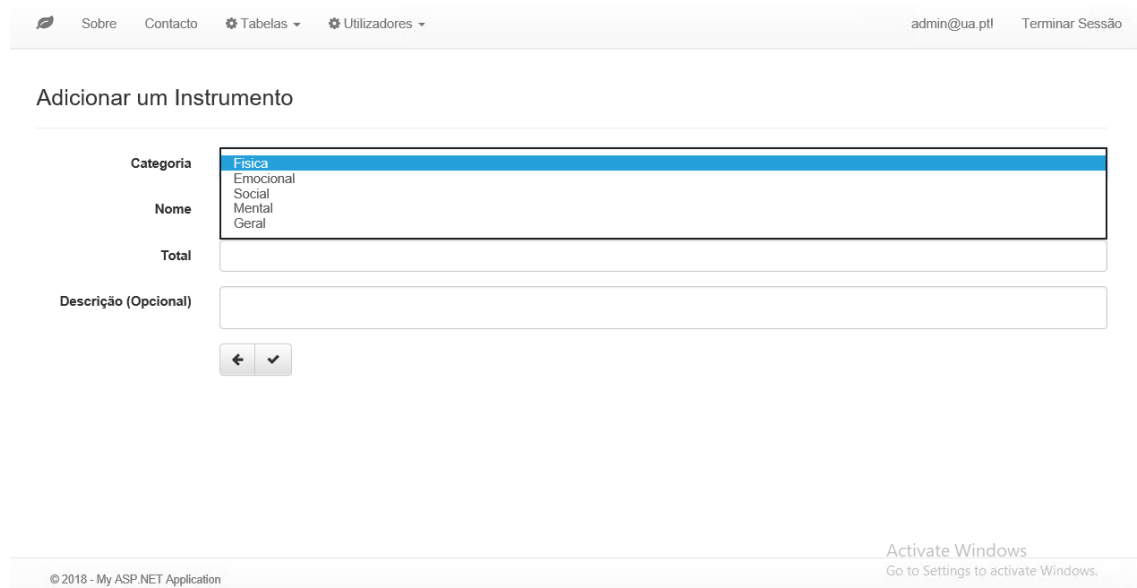


Figura 4.14: Página de criação de instrumentos

A página de criação de novos elementos é constituída pela denominação dos campos que fazem parte do objeto a adicionar. No caso específico da criação de um instrumento no sistema, apresentado na Figura 4.14, é indicado os dados a preencher em cada entrada, sendo eles: Categoria, que tem que ser seleccionada de uma lista que já contém as Categorias existentes no sistema, e entradas de texto para o Nome, Total e Descrição, que é um campo opcional.

Depois de adicionados, os novos elementos podem ser vistos em detalhe, editados ou removidos.

A página de edição dos elementos é semelhante à de criação, no entanto os campos previamente preenchidos/selecionados contêm as respostas submetidas anteriormente em modo editável, permitindo assim a adição, correção ou alteração de informação por parte do administrador. Por outro lado, a página de remoção de elementos irá exibir a informação do elemento e pedir confirmação da eliminação do mesmo, prevenindo que a remoção de um elemento pode envolver a remoção de outros elementos dele dependentes. Um exemplo disto é exibido na Figura 4.15 que apresenta a informação importante do instrumento e avisa que ao eliminá-lo, o administrador do sistema estará também a eliminar as questões que dele dependem.

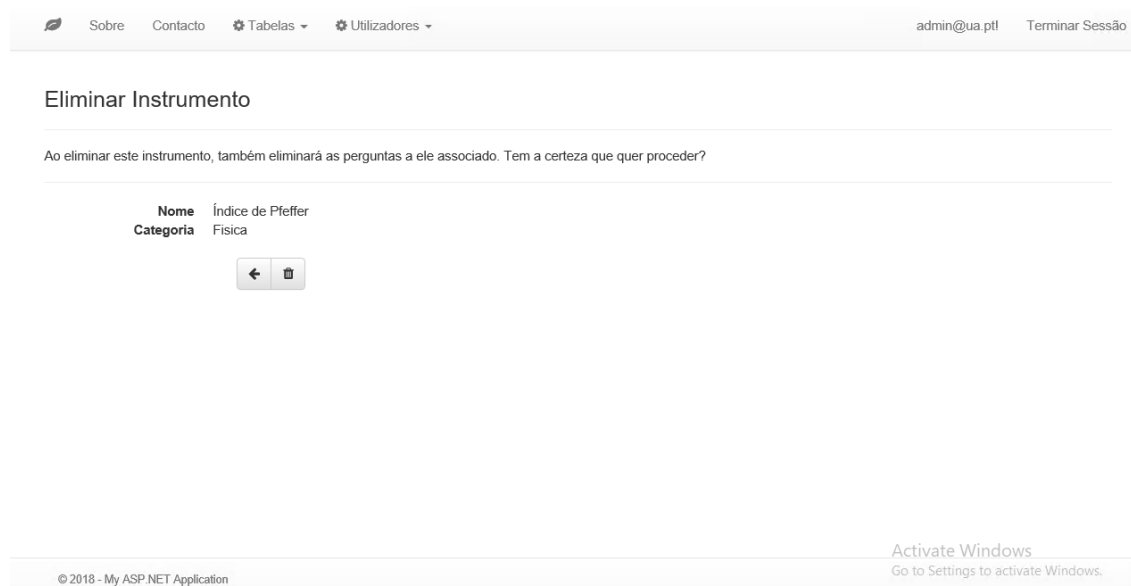


Figura 4.15: Página de eliminação do instrumento Índice de Pfeffer

Por fim, a última página em comum é a de "Ver Detalhes". Esta página exibe todos os campos que pertencem a um dado elemento. Existe também, como em todas as outras páginas, um ícone que permite voltar à página anterior. Em adição, a View dos detalhes inclui um ícone adicional que direciona para a edição do elemento.

A View de detalhes de um instrumento, no entanto, tem funcionalidade adicionais em relação às restantes views dos diversos elementos, uma vez que permite fazer a gestão das questões associadas ao instrumento no fim da página. A Figura 4.16 contém a página apresentada no caso de ainda não existirem questões associadas a um instrumento, enquanto que na Figura 4.17 é exibida a página de detalhes da Escala de Barthel, que inclui uma lista das suas questões e o direcionamento tanto para uma página de criação de uma nova questão neste instrumento (Figura 4.18) como para a página de edição ou remoção de uma questão em específico.

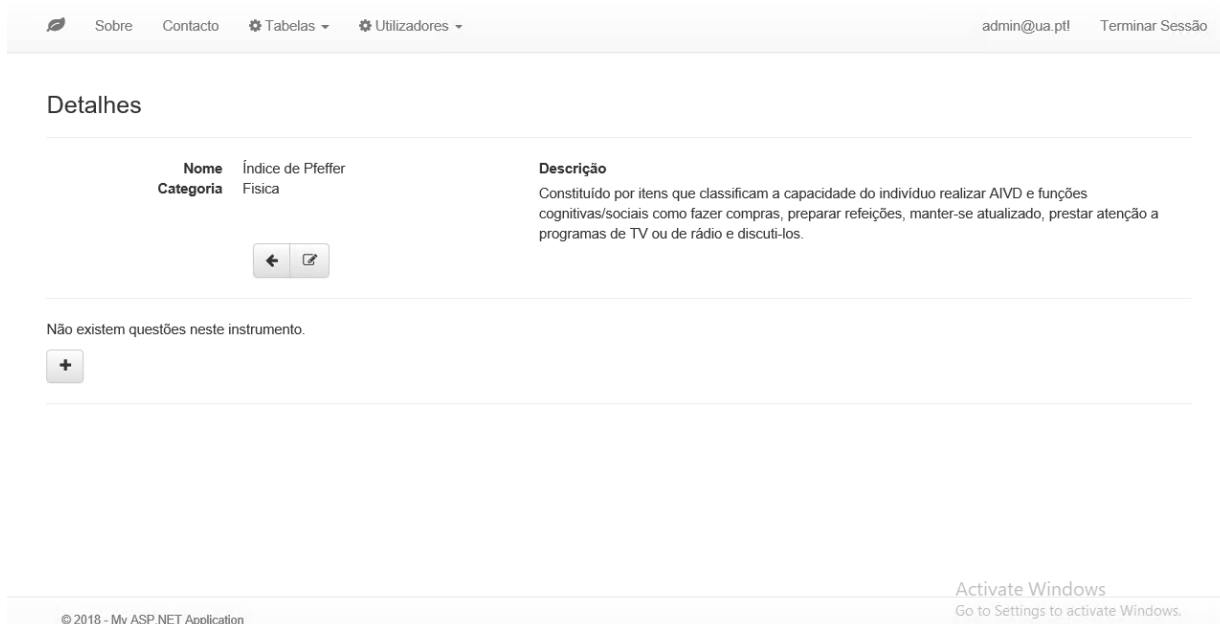


Figura 4.16: Página "Ver Detalhes" do instrumento Índice de Pfeffer

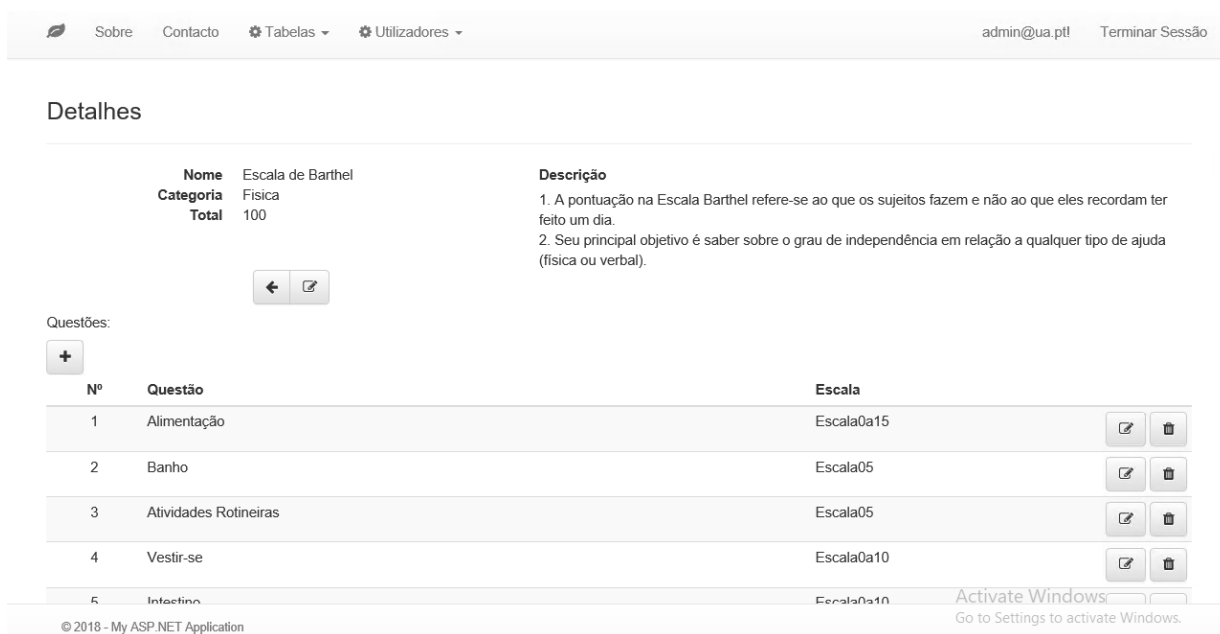


Figura 4.17: Página "Ver Detalhes" do instrumento Escala de Barthel

Sobre Contacto Tabelas Utilizadores admin@ua.pt! Terminar Sessão

Adicionar uma questão

Instrumento Índice de Pfeffer

Tipo de Resposta Escala01

Posição

Questão

← ✓

© 2018 - My ASP.NET Application

Activate Windows
Go to Settings to activate Windows.

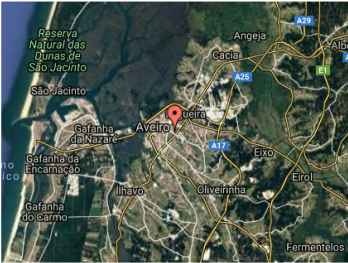
Figura 4.18: Página de criação de uma nova questão do Índice de Pfeffer

Outra View de "Ver Detalhes" que contém funcionalidades próprias é a relativa às instituições (Figura 4.19). Caso a morada inserida seja válida, é possível ter uma visualização da localização da instituição num mapa e explorar a partir daí.

HOME Sobre Contacto Tabelas Utilizadores admin@ua.pt! Terminar Sessão

Detalhes

Nome	Hospital da Luz
Endereço	Rua do Brasil n21
Código Postal	3800-009
Cidade	Aveiro
Latitude	40,6375999
Longitude	-8,6363764



← ✓

© 2018 - My ASP.NET Application

Activate Windows
Go to Settings to activate Windows.

22:13
27/09/2018

Figura 4.19: Página de detalhes da instituição Hospital da Luz Aveiro

Views Utilizador

No caso do utilizador comum, a sua barra de navegação será composta por ligações às páginas "Home", "Sobre" e "Contactos" inicial da plataforma, e duas dropdowns: uma relativa às avaliações e outra aos utentes, como se pode ver na Figura 4.20. Acresce-se o menu da esquerda, que permite ir para a parte de gestão da sua própria conta ao carregar no seu email ou terminar sessão.

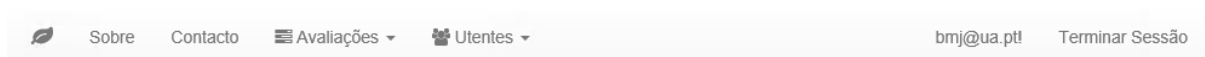


Figura 4.20: Barra de navegação do utilizador

O menu "Avaliações", presente na Figura 4.21, contém duas opções: ir para a página de criação de uma nova avaliação ou ver as avaliações já submetidas no sistema. Por outro lado, o menu "Utentes", representado na Figura 4.22, permite ir para a página de criação de um novo utente ou ver a lista de utentes já existentes.

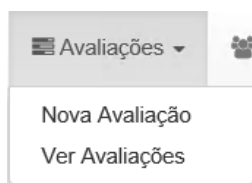


Figura 4.21: Menu de Gestão de Avaliações

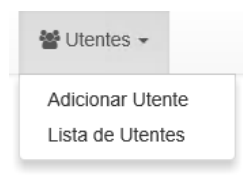


Figura 4.22: Menu de Gestão de Utentes

As views relativas às listas tanto da Gestão de Utentes como da Gestão de Avaliações seguem o formato descrito na lista de elementos nas páginas do administrador. O mesmo acontece no caso da remoção e edição das mesmas, no entanto, o processo de criação de uma avaliação e a visualização de detalhes de utentes e avaliações acresce funcionalidades específicas.

No caso do registo de uma nova avaliação no sistema, apresentado na Figura 4.23, é necessário seleccionar uma data válida e preencher todos os campos obrigatórios. Depois de clicar no ícone para continuar, será apresentado o conteúdo do instrumento, isto é, as questões ou itens nele incluídos e as respostas possíveis para cada questão, como se pode ver na Figura 4.24.

[Sobre](#)
[Contacto](#)
[Avaliações](#)
[Utentes](#)

bmj@ua.pt
Terminar Sessão

Detalhes da Avaliação

Data

Data de Catalogação 27/09/2018

Data de realização 15/01/2018

Instrumento Utilizado

Instrumento Escala de Barthel

Categoria Fisica

Instituição

Instituição Hospital da Luz

Prestador de Serviços

Nome Completo Barbara Miguel Junho

E-mail bmj@ua.pt

Utente

Nome Completo Joana Morais de Silva Pereira

Data de Nascimento 22/12/1945

Sexo F

Telefone/Telemóvel 268999999

←

✎

Figura 4.25: Página de detalhes de uma avaliação parte 1

uma secção de "Questões e Respetivas Respostas", Figura 4.26, que apresenta uma lista de todas as questões e respetivas respostas, bem como o total obtido na avaliação. No caso de a avaliação ter sido executada e registada pelo utilizador em sessão, como é o caso, existem ainda botões de edição e remoção de cada uma das respostas. No caso de o utilizador não ser responsável pela avaliação, este só poderá consultar as respostas, não lhe será permitido edição nem remoção das mesmas.

Questões e Respetivas respostas			
Nº	Questão	Resposta	
1	Alimentação	15	
2	Banho	6	
3	Atividades Rotineiras	6	
4	Vestir-se	10	
5	Intestino	6	
6	Sistema Urinário	6	
7	Uso do Tóilet	10	
8	Transferência (da cama para a cadeira e vice-versa)	15	
9	Mobilidade (em superfícies planas)	15	
10	Escadas	6	
Total		95 / 100	Activate Windows

Figura 4.26: Página de detalhes de uma avaliação parte 2

Por fim, a última View que difere do padrão é a referente aos detalhes do utente, uma vez que esta serve como dashboard interativo que permite fazer um resumo e consultar

todas as informações relativas quer ao utente, quer às avaliações a ele realizadas.

Detalhes

Dados pessoais

Nome Completo: Joana Morais de Silva Pereira
Data de Nascimento: 22/12/1945
Sexo: F
Escolaridade: 9ºano

Contacto

Telefone/Telemóvel: 268999999

Avaliações do paciente

Data	Instrumento	Total	Total do Instrumento
15/01/2018 00:00:00	Escala de Barthel	90	100
09/05/2018 00:00:00	Escala de Barthel	70	100
01/06/2018 00:00:00	Escala de Barthel	50	100
03/07/2018 00:00:00	Escala de Barthel	65	100
04/07/2018 00:00:00	Patient Health Questionnaire (PHQ-9)	27	27
02/09/2018 00:00:00	Patient Health Questionnaire (PHQ-9)	25	27
16/09/2018 00:00:00	Escala de Barthel	60	100
27/09/2018 00:00:00	Patient Health Questionnaire (PHQ-9)	21	27

Figura 4.27: Detalhes de um utente - Dados pessoais, Contacto e Avaliações efetuadas

A título exemplificativo, pode-se considerar a utente Joana Morais da Silva Pereira, cuja página de detalhes inclui: a sua informação pessoal como utente do sistema, incluindo os seus dados e o contacto disponível (presente na Figura 4.27); lista de todas as avaliações executadas, com informações como o nome do instrumento utilizado e o total da avaliação comparativo com o total tabelado (também na Figura 4.27); e, por fim, as estatísticas do utente por instrumento, que consiste num gráfico por instrumento de pontuações atribuídas ao utente, ordenado por datas de modo a permitir visualizar caso exista incremento ou decréscimo significativo.

Na Figura 4.27, é apresentado o resto da página de detalhes da utente, que consiste na apresentação de um gráfico por instrumento, caso este tenha um total. Estes gráficos relacionam a data de realização da avaliação com a pontuação obtida pelo utente. A atualização de gráficos é dinâmica, na medida em que é atualizado mediante a criação de uma nova avaliação, e se outro instrumento com pontuações totais for utilizado na avaliação do utente em questão, um novo gráfico será criado automaticamente nesta página. Isto permite um melhor controlo e visualização dos resultados de cada utente.

02/09/2018 00:00:00	Patient Health Questionnaire (PHQ-9)	25	27	
16/09/2018 00:00:00	Escala de Barthel	60	100	
27/09/2018 00:00:00	Patient Health Questionnaire (PHQ-9)	21	27	

Estatísticas relativas às avaliações do utente por instrumento

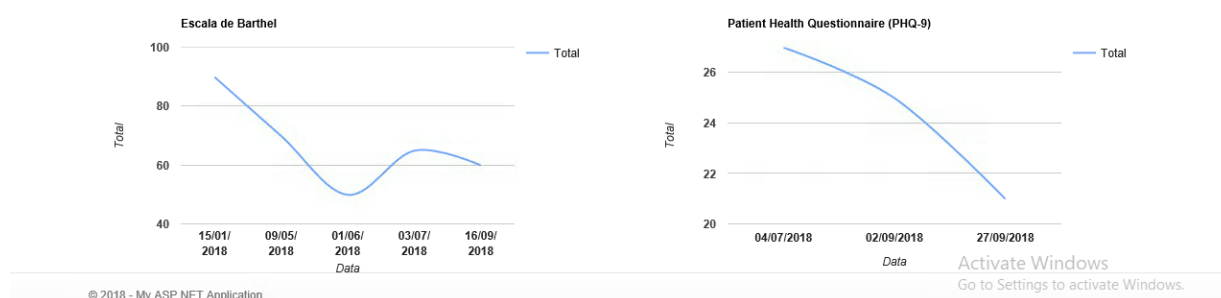


Figura 4.28: Detalhes de um utente - Estatísticas do utente por instrumento

```
function drawChart() {
    @(
    int x = 0;
    foreach (var item in Model.entryInfo)
    {
        if (item.instrument.total != null)
        {
            x++;
            string valores = "";
            foreach (var index in item.entries.OrderBy(t => t.dataReal))
            {
                valores += @Html.Raw(String.Format("{0:dd/MM/yyyy}", {1}), ", ", @index.dataReal, (@index.total ?? 0));
            }
            @Html.Raw("console.log(\"passei aqui 3\");\n");
            @Html.Raw("data" + @x + " = google.visualization.arrayToDataTable([['Data', 'Total'], " + @valores + "]);\nconsole.log(data" + @x + ");\n");
        }
    }
    )
    var options = {
        title: "",
        curveType: 'function',
        legend: { position: 'right' },
        hAxis: { title: 'Data', textStyle: { fontName: 'Arial', bold: true, italic: false } },
        vAxis: { title: 'Total', textStyle: { color: '#FFFFFF', bold: true, italic: false } },
        colors: ['#76A7FA']
    };
    @(
    int i = 1;
    foreach (var item in Model.entryInfo)
    {
        if (item.instrument.total != null)
        {
            @Html.Raw("var chart" + @i + " = new google.visualization.LineChart(document.getElementById('curve_chart' + @i + ''))");
            @Html.Raw("chart" + @i + ".draw(data" + @i + ", { title: \" " + item.instrument.nome + "\", curveType: 'function', legend: { position: 'right' }, " +
            "hAxis: { title: 'Data', textStyle: { fontName: 'Arial', bold: true, italic: false } }, " +
            "vAxis: { title: 'Total', textStyle: { color: '#FFFFFF', bold: true, italic: false } }, colors: ['#76A7FA']});");
            i++;
        }
    }
    )
}
```

Activate Wind
Go to Settings to a

Figura 4.29: Código javascript e html com C Sharp embutido para criação dinâmica de um gráfico

Um dos maiores desafios do desenvolvimento desta plataforma foi a criação de gráficos

que apresentem uma relação entre os resultados das várias avaliações pertencentes ao mesmo instrumento. Para este efeito, foi necessário criar um Modelo com classes que contém o instrumento e a lista de avaliações por utente. Estas classes já foram apresentadas anteriormente, na Figura 4.5, e utilizando javascript, HTML e código C Sharp embutido, é possível ir buscar as avaliações do paciente em questão para cada um dos instrumentos. O principal objetivo consiste em desenhar um gráfico que relaciona a data de realização das diferentes avaliações com os respetivos resultados totais, permitindo assim se houve alterações significativas ao longo do tempo.

Na Figura 4.29, é apresentado o código que permite o desenho dos gráficos em questão. Foi utilizado o Google Charts para visualização de dados em formato de gráfico de curvas [58]. Para cada uma das avaliações, verifica-se se o instrumento contém um total. Caso esta condição se verifique, as diferentes entradas são ordenadas por data e os campos data de realização e total da avaliação tornam-se os eixos x e y do gráfico, respetivamente. Num ciclo, adiciona-se os campos data de realização e total de cada uma das avaliações. Depois, define-se as opções relativas à aparência do gráfico e este é desenhado. Existem instrumentos que não possuem total e, assim sendo, os seus resultados não são apresentados em formato de gráfico, no entanto, as avaliações estarão organizadas na lista de avaliações efetuadas por data e podem ser consultadas de forma fácil e interativa.

4.4.5 Integração

Da integração dos três componentes descritos anteriormente (Modelo, Controlador e View) resulta um sistema que atua como suporte à avaliação multidimensional da funcionalidade de pessoas idosas. Este sistema permite então a gestão de utilizadores e de instrumentos, bem como todos os seus componentes, por parte de um administrador e a gestão e avaliação de utentes por prestadores de cuidados que podem ter diversos papéis. Para além disto, é possível que o prestador de cuidados encarregado de um certo utente aceda à sua página e veja, de forma simples e fácil de usar, os dados relativos a todas as avaliações que foram realizadas a esse utente, de modo a obter uma perspetiva geral e detetar anomalias e deterioração do estado da pessoa idosa.

Capítulo 5

Conclusão e Trabalho Futuro

Nesta secção serão apresentadas as conclusões e sugestões para trabalho futuro no âmbito desta dissertação.

5.1 Conclusão

Apesar de o tema da avaliação multidimensional da funcionalidade de pessoas idosas ser bastante atual, ainda não existe um consenso a nível da aplicação de instrumentos de avaliação normalizados. No entanto, existe um conjunto muito diversificado de instrumentos para avaliação das diferentes dimensões de funcionalidade, que são utilizados por prestadores de cuidados de saúde e de âmbito social.

No caso de uma pessoa idosa, a questão da funcionalidade e qualidade de vida pode influenciar gravemente a sua saúde, o que torna a sua monitorização e controlo ainda mais importantes.

O trabalho desta dissertação foi desenvolvido no sentido de poder ser utilizado no suporte à avaliação multidimensional da funcionalidade de pessoas idosas e o seu uso poderia ser uma mais valia na vida destas pessoas. O uso contínuo de instrumentos adequados e sua análise ao longo do tempo pode permitir aos prestadores de cuidados perceberem anomalias e tomarem decisões e intervirem quando necessário.

Já existem diversos sistemas de inquéritos que apresentam um leque alargado de modelos disponíveis e, caso nenhum destes for adequado, disponibiliza ferramentas de personalização de inquéritos à medida do cliente. De um modo geral, estes sistemas permitem fazer questionários muito completos, mas não têm funcionalidades suficientes para serem usados na avaliação multidimensional da funcionalidade de pessoas idosas, uma vez que esta avaliação é multi-funcional e dividida em diferentes categorias e que o sistema tem que suportar o armazenamento da informação tanto de prestadores de cuidados, como de utentes.

Existem também diversos instrumentos que permitem avaliar a funcionalidade de pessoas idosas, tanto a um nível global como numa dimensão específica. Numa era tão digital como a atualidade, existe uma necessidade inerente de digitalizar também os diversos ins-

trumentos utilizados pelos prestadores de cuidados no sentido de avaliar a funcionalidade de pessoas idosas, de modo a otimizar a prestação de cuidados.

O sistema de informação desenvolvido no contexto desta dissertação atingiu os objetivos propostos e cumpre todos os requisitos, funcionais e não funcionais, inicialmente propostos.

5.2 Trabalho Futuro

Apesar de o sistema de informação desenvolvido no contexto desta dissertação já apresentar todos os componentes essenciais aos objetivos propostos, há espaço para melhorias e funcionalidades adicionais adaptadas às necessidades dos prestadores de cuidados.

A nível dos utilizadores, existem diversos profissionais no ramo da saúde com competências que divergem. Tendo em conta o Regulamento Geral de Proteção de Dados que entrou em vigor em 25 de Maio de 2018, no futuro é imprescindível a adição de vários papéis de utilizador: médico familiar, médico hospitalar, enfermeiro, psicólogo, fisioterapeuta, cuidador de âmbito social ou cuidador informal, entre outros. Assim, os utilizadores só teriam acesso aos instrumentos da sua área específica e, conseqüentemente, existirá uma hierarquia entre os mesmos. Assim, utilizadores com diferentes papéis teriam diferentes níveis de acesso e garante-se a proteção dos dados.

Outra melhoria a ter em conta, seria a integração de um novo tipo de utilizador, o paciente. Já existem diversos exercícios a nível cognitivo que só ajudariam na avaliação do estado da funcionalidade do idoso se fossem preenchidos pelo próprio. Teriam de ser exercícios adequados e ter em mente que os utentes, sendo pessoas idosas que muitas vezes não sabem utilizar um computador, poderiam precisar da ajuda de prestadores de cuidados ou familiares para o seu preenchimento.

O sistema desenvolvido já responde à maior parte das necessidades de um sistema com este objetivo, mas fica a ideia que ainda é necessário trabalho adicional.

Bibliografia

- [1] K. Kroenke, R. L. Spitzer, and J. B. Williams, “The PHQ-9: validity of a brief depression severity measure.” *Journal of general internal medicine*, vol. 16, no. 9, pp. 606–13, sep 2001. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11556941><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1495268>
- [2] Z. S. Nasreddine, N. A. Phillips, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings, and H. Chertkow, “The Montreal Cognitive Assessment, MoCA: A Brief Screening Tool For Mild Cognitive Impairment,” *Journal of the American Geriatrics Society*, vol. 53, no. 4, pp. 695–699, apr 2005. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/15817019><http://doi.wiley.com/10.1111/j.1532-5415.2005.53221.x>
- [3] “Survio.” [Online]. Available: <https://www.survio.com/>
- [4] “SurveyMonkey.” [Online]. Available: <https://pt.surveymonkey.com/>
- [5] “QuestionPro.” [Online]. Available: <https://www.questionpro.com/>
- [6] “Introduction to Oracle Database.” [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/18/cncpt/introduction-to-oracle-database.html{\#GUID-8F2EEEC8-0372-4419-88FF-7D77A9C0FCAD}>
- [7] A. Syromiatnikov and D. Weyns, “A Journey through the Land of Model-View-Design Patterns,” in *2014 IEEE/IFIP Conference on Software Architecture*. IEEE, apr 2014, pp. 21–30. [Online]. Available: <http://ieeexplore.ieee.org/document/6827095/>
- [8] N. George, *Mastering Django: Core*. Packt Publishing, 2016. [Online]. Available: <https://www.packtpub.com/application-development/mastering-django-core>
- [9] Instituto Nacional de Estatística, “Projeções de População Residente 2015-2080,” *Instituto Nacional de Estatística*, no. 2015, pp. 1–19, 2017.
- [10] “The World Health Organization Quality of Life Assessment (WHOQOL): development and general psychometric properties.” *Social science & medicine* (1982), vol. 46, no. 12, pp. 1569–85, jun 1998. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/9672396>

- [11] “Relatório Mundial de Envelhecimento e Saúde.” World Health Organization, Tech. Rep., 2015. [Online]. Available: www.who.int
- [12] B. Brorsson and K. H. Asberg, “Katz index of independence in ADL. Reliability and validity in short-term care.” *Scandinavian journal of rehabilitation medicine*, vol. 16, no. 3, pp. 125–32, 1984. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/6494836>
- [13] D. T. Wade and C. Collin, “The Barthel ADL Index: a standard measure of physical disability?” *International disability studies*, vol. 10, no. 2, pp. 64–7, 1988. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/3042746>
- [14] M. Lawton and E. Brody, “Assessment of older people: Self-maintaining and instrumental activities of daily living,” Tech. Rep. 1, 2006. [Online]. Available: www.hartfordign.organd/orwww.ConsultGerIRN.org
- [15] R. I. Pfeffer, T. T. Kurosaki, J. M. Chance, S. Filos, and D. Bates, “Use of the mental function index in older adults. Reliability, validity, and measurement of change over time.” *American journal of epidemiology*, vol. 120, no. 6, pp. 922–35, dec 1984. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/6507430>
- [16] J. H. Ansai, S. F. N. Glisoi, T. Oliveira, A. T. Soares, K. N. Cabral, C. Sera, and S. P. Paschoal, “Revisão de dois instrumentos clínicos de avaliação para prever risco de quedas em idosos,” *Revista Brasileira de Geriatria e Gerontologia*, vol. 17, no. 1, pp. 177–189, mar 2014. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1809-98232014000100177&lng=pt&tlng=pt
- [17] T. M. Steffen, T. A. Hacker, and L. Mollinger, “Age- and gender-related test performance in community-dwelling elderly people: Six-Minute Walk Test, Berg Balance Scale, Timed Up & Go Test, and gait speeds.” *Physical therapy*, vol. 82, no. 2, pp. 128–37, feb 2002. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11856064>
- [18] C. S. Sterke, S. L. Huisman, E. van Beeck, C. W. N. Looman, and T. J. M. van der Cammen, “Is the Tinetti Performance Oriented Mobility Assessment (POMA) a feasible and valid predictor of short-term fall risk in nursing home residents with dementia?” *International Psychogeriatrics*, vol. 22, no. 02, p. 254, mar 2010. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19951457http://www.journals.cambridge.org/abstract/S1041610209991347>
- [19] R. Schwendimann, S. De Geest, and K. Milisen, “Evaluation of the Morse Fall Scale in hospitalised patients,” *Age and Ageing*, vol. 35, no. 3, pp. 311–313, may 2006. [Online]. Available: <http://academic.oup.com/ageing/article/35/3/311/40246/Evaluation-of-the-Morse-Fall-Scale-in-hospitalised>

- [20] I. Santana, D. Duro, R. Lemos, V. Costa, M. Pereira, M. R. Simões, and S. Freitas, “Mini-mental state examination: Avaliação dos novos dados normativos no rastreio e diagnóstico do défice cognitivo,” *Acta Medica Portuguesa*, vol. 29, no. 4, pp. 240–248, 2016.
- [21] E. Pfeiffer, “A short portable mental status questionnaire for the assessment of organic brain deficit in elderly patients.” *Journal of the American Geriatrics Society*, vol. 23, no. 10, pp. 433–41, oct 1975. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/1159263>
- [22] R. M. Oliveira, “O Teste do Relógio: tempo de mudança?” pp. 1–54, 2013.
- [23] Z. Shao, E. Janse, K. Visser, and A. S. Meyer, “What do verbal fluency tasks measure? Predictors of verbal fluency performance in older adults.” *Frontiers in psychology*, vol. 5, p. 772, 2014. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/25101034><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4106453>
- [24] M. Duque, A.S., Gruner, H., Clara, J.G., Ermida, J.G. and Veríssimo, “Avaliação Geriátrica,” *Núcleo de Estudos de Geriatria da Sociedade Portuguesa de Medicina Interna (GERMI)*, pp. 1–20, 2012. [Online]. Available: http://www.spmi.pt/docs{_}nucleos/GERMI{_}36.pdf.
- [25] K. Kroenke, R. L. Spitzer, J. Williams, and B. Löwe, “The Patient Health Questionnaire Somatic, Anxiety, and Depressive Symptom Scales: a systematic review,” *General Hospital Psychiatry*, vol. 32, no. 4, pp. 345–359, jul 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0163834310000563?via{\%}3Dihub>
- [26] O. Ribeiro, L. Teixeira, N. Duarte, M. J. Azevedo, L. Araújo, S. Barbosa, and C. Paúl, “Versão Portuguesa da Escala Breve de Redes Sociais de Lubben (LSNS-6),” *Revista Temática Kairós Gerontologia*, vol. 15, no. 1, pp. 217–234, 2012.
- [27] J. L. Pais-Ribeiro and A. C. S. Ponte, “Propriedades métricas da versão portuguesa da escala de suporte social do Mos (MOS Social Support Survey) com idosos,” pp. 163–174, 2009.
- [28] J. Ribeiro, “Escala de satisfação com o suporte social (ESSS),” *Análise psicológica*, pp. 1–25, 2011.
- [29] B. B. Hamilton, J. A. Laughlin, R. C. Fiedler, and C. V. Granger, “Interrater reliability of the 7-level functional independence measure (FIM),” *Scandinavian journal of rehabilitation medicine*, vol. 26, no. 3, pp. 115–9, sep 1994. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/7801060>
- [30] J. E. Ware and C. D. Sherbourne, “The MOS 36-item short-form health survey (SF-36). I. Conceptual framework and item selection.” *Medical care*, vol. 30, no. 6, pp. 473–83, jun 1992. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/1593914>

- [31] “World Health Organization,” *The world health organization quality of life (WHOQOL)-BREF*, 2004. [Online]. Available: <http://www.who.int/substance{\\-}abuse/research{\\-}tools/whoqolbref/en/>
- [32] R. M. C. Rodrigues, “Validação da versão em português europeu de questionário de avaliação funcional multidimensional de idosos,” *Revista Panamericana de Salud Pública*, vol. 23, no. 2, pp. 109–115, 2008. [Online]. Available: <http://www.scielo.org/scielo.php?script=sci{\\-}arttext{\\&}pid=S1020-49892008000200006{\\&}lng=pt{\\&}nrm=iso{\\&}tlng=pt>
- [33] T. Üstün, S. Chatterji, N. Kostanjsek, J. Rehm, C. Kennedy, J. Epping-Jordan, S. Saxena, M. von Korf, and C. Pull, “Developing the World Health Organization Disability Assessment Schedule 2.0,” *Bulletin of the World Health Organization*, vol. 88, no. 11, pp. 815–823, nov 2010. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/21076562http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2971503http://www.who.int/bulletin/volumes/88/11/09-067231.pdf>
- [34] W. group on health outcomes for older persons with multiple chronic Working Group on Health Outcomes for Older Persons with Multiple Chronic Conditions, “Universal health outcome measures for older persons with multiple chronic conditions.” *Journal of the American Geriatrics Society*, vol. 60, no. 12, pp. 2333–41, dec 2012. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/23194184http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3521090>
- [35] “Agile Methodology,” pp. 3–5, 2017. [Online]. Available: <http://agilemethodology.org/>
- [36] C. Coronel and S. Morris, *Database systems : design, implementation, and management*. Cengage, 2018.
- [37] J. Homan, “Relational vs. non-relational databases: Which one is right for you? — Pluralsight,” 2014. [Online]. Available: <https://www.pluralsight.com/blog/software-development/relational-non-relational-databases>
- [38] M. Taveggia, C. Guyer, B. Hamilton, S. Stein, J. Hubbard, and C. Rabeler, “SQL Server Management Studio (SSMS) — Microsoft Docs,” 2017. [Online]. Available: <https://docs.microsoft.com/pt-pt/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017{\\#}sql-server-management-studio-components>
- [39] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, “NoSQL databases: Critical analysis and comparison,” *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pp. 293–299, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8284494/>

- [40] “What is MySQL?” 2017. [Online]. Available: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- [41] “The main features of MySQL,” pp. 9–10, 2016. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/features.html>
- [42] “SQL Server Documentation.” [Online]. Available: <https://docs.microsoft.com/pt-pt/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>
- [43] C. Gyorodi, R. Gyorodi, G. Pecherle, and A. Olah, “A comparative study: MongoDB vs. MySQL,” *2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015*, pp. 0–5, 2015.
- [44] K. Chodorow, *MongoDB: The Definitive Guide*. O’Reilly Media, Inc, 2013. [Online]. Available: <http://usuaris.tinet.cat/bertolin/pdfs/mongodb{\-}thedefinitiveguide-kristinachodorow{\-}1401.pdf>
- [45] S. Smith, “Overview of ASP.NET Core MVC.” [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-2.1>
- [46] A. Syromiatnikov and D. Weyns, “A Journey through the Land of Model-View-Design Patterns,” in *2014 IEEE/IFIP Conference on Software Architecture*. IEEE, apr 2014, pp. 21–30. [Online]. Available: <http://ieeexplore.ieee.org/document/6827095/>
- [47] “ASP.NET Overview.” [Online]. Available: <https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>
- [48] D. S. Foundation, “Django Documentation,” pp. 1–1172, 2012. [Online]. Available: <http://media.readthedocs.org/pdf/django/1.4.X/django.pdf{\%}5Cnpapers2://publication/uuid/C3107E85-413D-4CD9-828F-B9BAFB777EF0>
- [49] K. Chowdhury, *Mastering Visual Studio 2017*. Packt Publishing, 2017. [Online]. Available: <https://www.oreilly.com/library/view/mastering-visual-studio/9781787281905/>
- [50] M. Escalona and N. Koch, “Requirements Engineering for Web Applications-A Comparative Study,” *Journal of Web Engineering*, vol. 2, no. 3, pp. 193–212, 2004. [Online]. Available: <http://www.pst.informatik.uni-muenchen.de/{~}kochn/KochEscalonaJWE-rev.pdf>
- [51] “Bootstrap.” [Online]. Available: <https://getbootstrap.com/docs/3.3/>
- [52] “jQuery.” [Online]. Available: <https://jquery.com/>
- [53] “Object Relational Mapper.” [Online]. Available: <https://www.devmedia.com.br/orm-object-relational-mapper/19056>

- [54] D. K. Barry, “Transparent Persistence vs. JDBC Call-Level Interfaces.” [Online]. Available: https://www.service-architecture.com/articles/object-relational-mapping/transparent{_}persistence{_}vs{_}jdbc{_}call-level{_}interface.html
- [55] J. Berkus, “Wrecking Your Database.” [Online]. Available: <https://it.toolbox.com/blogs/josh-berkus/wrecking-your-database-080509>
- [56] “Code First vs Database First.” [Online]. Available: <https://entityframework.net/code-first-vs-database-first>
- [57] T. FitzMacken, “Getting Started with Entity Framework 6 Database First using MVC 5 — Microsoft Docs,” 2014. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/database-first-development/setting-up-database>
- [58] “Google Charts.” [Online]. Available: <https://developers.google.com/chart/interactive/docs/>